

MARCH 1992
VOLUME 5, NO. 3

\$3.95
\$4.95 Canadian

DATABASE

Programming & Design

**Choosing the
Right Client for
Your Server**

**Building a Stable
(But Flexible)
Relational Design**

**Distributed
Database: Where
Things Stand Now**

**Downsizing the
Corporate Database:
A Primer**



MARCH 1992
VOLUME 5, NO. 3

\$3.95
\$4.95 Canadian

DATABASE

Programming & Design

**Choosing the
Right Client for
Your Server**

**Building a Stable
(But Flexible)
Relational Design**



MIRACLES DO HAPPEN

See for yourself...
come to Information Builders.
Booth #818

How to get to your corporate data easily.

If you don't know the first thing about SQL, you already know the first thing about Quest.[™]

Can you use Microsoft[®] Windows[™]? Then you can use Quest.[™] Quest's menus, windows and icons make it fast and easy to unlock a treasure chest of corporate information without any knowledge of SQL.

Use Quest to access DB2, Oracle[®], OS/2[®] Extended Edition Database Manager or your favorite SQL database server.

Then, use your newly acquired data anyway you want. Prepare reports the easy way, with Quest. Crunch the data with Excel or Lotus 1-2-3.[®] Make it look pretty with PageMaker.[®]

Best of all, you don't have to do any programming. You don't even need to know how to pronounce "SQL."

Your Windows and PC LAN database connection.

Quest is the graphical data access tool for everyone. There's nothing better than Quest for getting data from mainframes, minicomputers or PC LAN servers. But you can even use Quest without a server, on your own PC.

Suppose you want to use records in a dBase file for Microsoft Word mail merge. Solution: give your assistant Quest.

Your letters will be on your desk in no time, thanks to Quest's point-and-click simplicity.

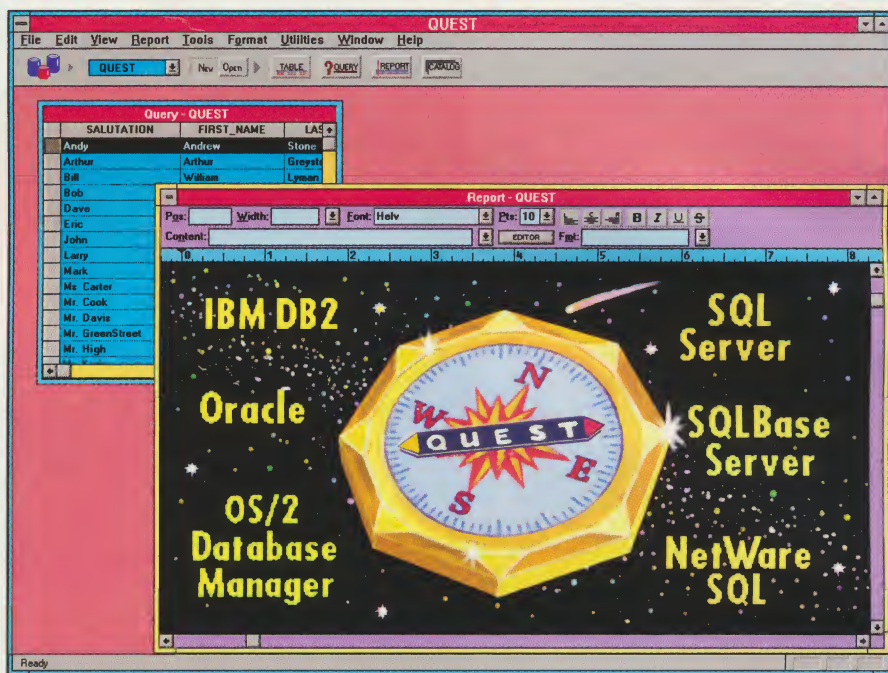
Use Quest with confidence.

Quest is the latest member of the Gupta SQL System, which includes SQLWindows[®], SQLNetwork[™] and our SQLBase[®] Server.

Ask your MIS department to install SQLNetwork, giving you safe access to corporate data stored in your company's mainframe or minicomputer database.

Use Quest with confidence. Gupta products have won rave reviews for years from experts and independent reviewers.

"Database Tool of the Year"
LAN Magazine (1989)
"Best SQL Product"
Data Based Advisor (1990)
"Best Front End Tool"
DBMS Magazine (1990)
"1990 Byte Award of Distinction"
Byte (1990)



Now, create sophisticated reports and queries with Quest.

Quest is the perfect companion to all your SQL applications. If you use database programs created with SQLWindows, you can use Quest to create ad hoc reports and queries without additional programming.

Take a look for yourself.

Now you can get a closer look at your personal Window to SQL.

Just send for your free Quest Demonstration Kit, complete with a Quest demo disk and the *Quest Solutions* booklet.

You'll learn how to create reports, graphs or tables, and arrange data on your PC any way you want. Because with Quest, what you see on the screen is what you get.

To use Quest, all you need is Microsoft Windows 3.0, a PC with 2mb memory and a hard disk. You can use all of Quest's powerful features on a single, stand alone PC.

The full Quest program is available from your dealer at a suggested retail price of \$495. Or, to find out what Quest can do for you, call us for your free demo kit. The *Quest Solutions* booklet is packed with ways to use Windows to unlock your treasure chest of corporate information. And the free demo disk lets you see just how easy Quest really is to use.

Free demo kit.

Includes *Quest Solutions* booklet and demo disk. To order, call toll free.

1-800-388-4550 ext. 114



Gupta

We open windows to SQL

Graphical Tools Database Servers SQL Connectivity

1040 Marsh Road, Menlo Park, CA 94025

(415) 321-9500, FAX (415) 321-5471, Gupta Europe 44-628-478333

SQLBase and SQLWindows are registered trademarks and Quest, SQL System and SQLNetwork are trademarks of Gupta Technologies, Inc. Microsoft is a registered trademark, and Windows is a trademark of Microsoft Corporation. Oracle is a registered trademark of Oracle Corporation. IBM and OS/2 are registered trademarks of IBM Corporation. Lotus and 1-2-3 are registered trademarks of Lotus Development. PageMaker is a registered trademark of Aldus Corporation. NetWare is a registered trademark of Novell.

CIRCLE 1 ON READER SERVICE CARD



We'll help you see your repository a little differently.

It's something every kid knows but doesn't like to admit: if you keep your toybox organized, you get to spend more time playing and less time looking for things. The only problem is that keeping a neat toybox seems to spoil the fun.

As a grown-up MIS professional, keeping information organized is the key to success. And it doesn't have to spoil the fun.

That's where we come in. We're BrownStone Solutions. We offer the most complete, intelligently architected, mature DB2 Data Dictionary and Administration solutions you can get. The kind that take care of you and your users. DB2 solutions that make your job easier, your people more productive and which enjoy the largest installed base of any products of their kind.

How the big kids see it.

Citicorp, Hewlett-Packard Co., Phoenix Mutual Life Insurance, Reader's Digest, Pacific Bell, ICI Americas, Hewitt Associates and the U.S. Air Force are just a few who have chosen to make BrownStone an important part of their systems development environment.

They made that decision for some very good reasons.

For example, the DataDictionary/Solution lets you define a methodology incorporating the use of a repository, along with life cycle management, change controls, CASE and other tools. You can get a wide range of choices and all the benefits of a repository-based solution today.

BrownStone's new release gives you an intuitive SAA/CUA compliant dialog. Without writing a line of code, you control the contents of the pull-down menus as well as windows and the help facility to guide users through their tasks.

The big picture.

Only BrownStone DB2 products make these kinds of "workbench controls" an inherent part of their architecture. The BrownStone DataDictionary/Solution understands projects, users, tools (including those you create), and an unencrypted, extensible E-R based repository. This lets you automatically update menus, enforce dictionary security and more.

In addition, a family of Administration Solutions help you implement and manage DB2 and IMS databases, and bridge information to and from IEW workstations.

These toolsets are fully integrated into the DataDictionary/Solution to take full advantage of its power.

Put your repository to work.

BrownStone Solutions helps you leverage your DB2 investment to make your environment more practical, powerful and productive — and ready for AD/Cycle.

Call us now for our free, information-packed brochure, "Putting Your Repository to Work" — plus a free checklist to help you evaluate any repository-based product you may be considering.

Except the one in the picture.

BrownStone
SOLUTIONS
Putting Your Repository to Work.

BrownStone Solutions
295 Madison Avenue
New York, New York 10017
1-800-627-7001

DB2 and AD/Cycle are trademarks of IBM Corp.
© Copyright 1990 BrownStone Solutions, Inc.

CIRCLE 2 ON READER SERVICE CARD

DEVELOP IT ONCE AND FOR ALL.

These days, it seems that almost everyone has a favorite Graphical User Interface (GUI). And most organizations also have non-graphical user interfaces running on block mode terminals, character mode terminals and PCs. Which usually means that developers must spend months rewriting each application for each incompatible system.

Unless the applications are built with Oracle® Tools.

An application developed with Oracle Tools automatically adapts to the native look and feel of the computer on which it runs. On Sun, IBM, DEC, HP, PCs, Macintosh and virtually any other computer. Even on character and block mode terminals. All without changing a single line of code.

Today, Oracle Tools like SQL*Forms and SQL*Menu work with Microsoft Windows and Presentation Manager. And they fully support Motif, Open Look, Macintosh, block mode and character mode.

So your applications can be deployed across all the computers in your organization. Your users can take full advantage of their GUI without having to be re-trained. And your programmers don't waste time recoding applications for each user interface.

Call us at 1-800-633-0553 Ext. 5756 and you'll receive the free Oracle Tools Information Kit, illustrating the full capabilities of Oracle Tools.

It'll show you how to solve any GUI mess once and for all.

ORACLE®

Software for people who can't predict the future.

©1992 Oracle Corporation. ORACLE, SQL*Forms and SQL*Menu are registered trademarks of Oracle Corporation. Macintosh is a trademark of Apple Computer, Inc. Open Look is a trademark of AT&T. Motif is a trademark of Open Software Foundation, Inc. Presentation Manager is a trademark of IBM, Inc. Windows is a trademark of Microsoft Corporation. All other trade names referenced are the trademark of the respective manufacturer. Call for hardware and software requirements. Outside the U.S.A. but within North America, call 1-800-668-8925 for product, service, and seminar information.

Open Look

Macintosh

Motif

Character Mode

Block Mode

DATABASE

Programming & Design

MARCH 1992

A Client for Your Server

30

RICHARD FINKELSTEIN

A thinking person's guide to choosing front-end tools for client/server databases.

Lions, Tigers, and Downsizing

39

HERBERT A. EDELSTEIN

Downsizing the corporate database is a tricky business, with many inherent "gotchas."

Distributed Database: Where Are We Now?

46

M. TAMER ÖZSU AND PATRICK VALDURIEZ

Distributed database technology has come a long way since its inception—for better or worse.

How and When to Use Surrogate Keys

55

DEBORAH L. BROOKS

Here's a technique for keeping design flexibility as you pin down stable primary keys.

Navigating the DBMS Labyrinth

58

STEVEN CANIANO

Our DBMS selection criteria conclude with a discussion of systems and operational issues.



DEPARTMENTS

EDITOR'S BUFFER

5

Tracking client/server's advance guard.

ACCESS PATH

11

Data sharing: the good, the bad, and the ugly.

DATABASE DESIGN

15

The trials and tribulations of learning to share.

DBA SHOPTALK

21

The ins and outs of data element naming.

CELKO ON SQL

25

Scheming with encoding mechanisms.

REPOSITORY REPORT

67

Version control: learning the ropes.

FOR MANAGERS ONLY

70

Rewriting database history.

BIRDS OF A FEATHER

73

A perfect CASE for users groups.

DATABASE LIBRARY

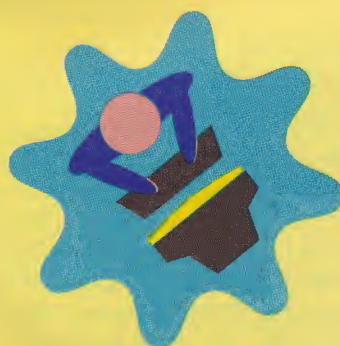
77

All you ever wanted to know about COBOL II.

PRODUCT WATCH

78

Marching to the beat of new products.



DATABASE PROGRAMMING & DESIGN (ISSN 0895-4518) is published monthly by Miller Freeman Publications, 600 Harrison St., San Francisco, CA 94107, (415) 905-2200. Please direct advertising and editorial inquiries to this address. For subscription inquiries, call (800) 289-0169 (outside U.S. (303) 447-9330). SUBSCRIPTION RATE for the U.S. is \$47 for 12 issues. Canadian/Mexican orders must be prepaid in U.S. funds with additional postage at \$6 per year. Canadian GST Permit #124513185. All other countries outside the U.S. must be prepaid in U.S. funds with additional postage at \$15 per year for surface mail or \$40 per year for air mail. POSTMASTER: Send address changes to DATABASE PROGRAMMING & DESIGN, P.O. Box 53481, Boulder, CO 80322-3481. For quickest service, call toll-free (800) 289-0169 (in Colorado or outside the U.S. (303) 447-9330). Please allow six weeks for change of address to take effect. SECOND CLASS POSTAGE paid at San Francisco, CA 94107 and at additional mailing offices. DATABASE PROGRAMMING & DESIGN is a registered trademark owned by the parent company, Miller Freeman Publications. All material published in DATABASE PROGRAMMING & DESIGN is copyrighted © 1991 by Miller Freeman Publications. All rights reserved. Reproduction of material appearing in DATABASE PROGRAMMING & DESIGN is forbidden without permission. 16mm microfilm, 35mm microfilm, 105mm microfiche and article and issue photocopies are available from University Microfilms International, 300 N. Zeeb Rd., Ann Arbor, MI 48106 (313) 761-4700.

DATABASE PROGRAMMING & DESIGN

THE ILLUMINATION YOU NEED TO UNLEASH THE HIDDEN POWER OF DB2.



Without the right tools, it is almost impossible to observe what's going on inside DB2. Inside the black box. As a result, tasks that should be being handled by various maintenance groups are consuming precious DBA resources.

That's why Candle offers a complete set of tools to illuminate DB2. Tools that observe problems based on your exception thresholds. Isolate the root cause of problems. And automatically fix problems or quickly recommend solutions.

Using Candle's tools, day-to-day Catalog

activities are simplified and automated. Overallocated DASD space is reclaimed. Time spent running reorgs and image copies is reduced. Poor-performing SQL is isolated and explained. And performance and service levels are easily monitored.

By simplifying and automating such time-consuming tasks, Candle's tools allow these tasks to be delegated to the appropriate groups, freeing the DBA to unleash the hidden power of DB2.

!Candle®



FREE ILLUMINATOR KIT

To become more enlightened about how Candle's powerful tools can streamline DB2 management and

administration, send for your free Illuminator Kit today. Complete the coupon and mail to Candle Corporation, 1999 Bundy Drive, Los Angeles, CA 90025. For faster service call 800 843-3970, Dept 415.

NAME	_____
COMPANY	_____
ADDRESS	_____
CITY	_____
STATE	_____
ZIP	_____

Copyright © 1991 Candle Corporation. All Rights Reserved

CIRCLE 3 ON READER SERVICE CARD

DB2 is a registered trademark of International Business Machines, Inc.

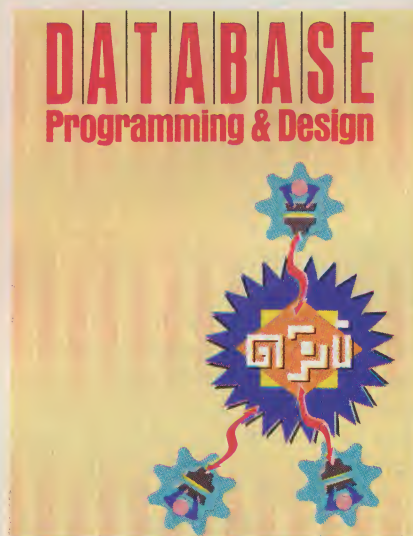
EDITOR'S BUFFER

Client/server computing forces us to reexamine the data-centric approach

AS HERB EDELSTEIN observes this month, there's much more to downsizing systems than kicking out the big computers and replacing them with smaller ones. Downsizing is just the most graphic of the various phrases (cooperative processing, client/server computing, open systems) being used to describe the trend toward decentralized, distributed computing: something that has many MIS shops and their favorite proprietary vendors trembling. It has brought us new enterprise concepts, such as IBM's Information Warehouse and Digital Equipment Corp.'s Network Application Support, and has made *heterogeneous* a household word. It is also opening up considerable new opportunities, hazards, and topics to reexamine.

In fact, the client/server architecture may force us to revisit the whole notion of a database management system. A DBMS, to use C.J. Date's definition, is "a computerized system whose overall purpose is to maintain information and to make that information available on demand." Date identifies the components of a database system as data, hardware, software, and users. The multiuser DBMS establishes a central point of control from which to govern the data resource requirements of related software (report writers, application tools, utilities, and so forth) plus the other three components. Since the relational revolution, the emphasis has been on managing the data. Software development complexity, the exponential increase in data, and the diffusion in types of users coming at the database has pushed the DBMS notion up to a higher level; and thus, we now talk of the repository, a facility to manage data about the data.

Return of the Process



But progress towards greater centralization is being outpaced (and perhaps undercut) by other developments. In a distributed environment, data, hardware, software, and users are dispersed. Responding to the crying need but slow pace of distributed database solutions, hardware and software vendors developed the notion of client/server computing. Few vendors still seem to be pursuing a truly data-centric solution to distributed database (the most notable survivor may be IBM, with its Distributed Relational Database Architecture). The client/server approach increasingly focuses on distributing *processes*.

The data warehouse, which was popularized by IBM last September (and covered extensively the past year by Bill Inmon in For

Managers Only), is a good example of the architectures springing up to help MIS shops figure out how to answer the need for process distribution, and yet control it. This approach splits applications into two categories, decision-support systems (DSS) and online transaction processing (OLTP). The DSS category rests more on the client side, while the OLTP applications more intensively involve the server. The distinction highlights the two demands tugging the database in the opposing directions: business flexibility and performance.

THE CLIENT/SERVER approach was at first hamstrung by the lack of "front ends": software applications designed to interact with a server, without requiring the user to write a lot of difficult code. But not anymore; as we see in Richard Finkelstein's article, DBMS-independent tools are proliferating the market. Attendees to some recent database conferences, including the upcoming DB/Expo (San Francisco, March 23 through 26) could certainly testify to the explosion of data access and retrieval software. Most such products now reach the database server by hooking into an application programming interface provided by the DBMS vendor (although perhaps someday a standardized interface will be built to industry specifications, such as SQL Access Group's).

As Finkelstein details, the front-end/back-end connection is still not smooth: facilities overlap, access languages are not always harmonious, and concurrency and integrity issues remain. Problems that may not have existed in a tight, proprietary relationship between tool and DBMS suddenly threaten. In the months ahead, we

DATABASE

Programming & Design

EDITOR

David Stodder

MANAGING EDITOR

Annalisa Chamberlain

EDITORIAL ASSISTANT

Theresa Rigney

TECHNICAL EDITORS

Howard Fosdick Marc Rettig

ART DIRECTOR

Mary Webster

COLUMNISTS

Joe Celko Terry Moriarty
William H. Inmon Jonathan Sayles
Barbara von Halle



NATIONAL SALES MANAGER/ ADVERTISING SALES STAFF

Chris Keating
Jamie Claar
East (212) 683-9294

REGIONAL ADVERTISING MANAGER/ ADVERTISING SALES STAFF

John Murphy
Juanita Nessinger
West (415) 905-2344

PUBLISHER

Steve Schneiderman

ASSOCIATE PUBLISHER

Phillip Chapnick

GROUP PUBLISHER

Donald Pazour

PRODUCTION COORDINATOR

Ron Bertola

DIRECTOR OF PRODUCTION

Andrew A. Mickus

DIRECTOR OF CIRCULATION

Jerry M. Okabe

CIRCULATION MARKETING MANAGER

Kathy E. Henry

NEWSSTAND MANAGER

Pam Santoro

ADMINISTRATIVE MANAGER

Joyce Schroeder



CHAIRMAN OF THE BOARD

Graham J. S. Wilson

PRESIDENT/CEO

Marshall W. Freeman

SENIOR VICE PRESIDENTS

Thomas L. Kemp Steven G. Stoneburn
Daniel J. Mills Wini D. Ragus
H. Verne Packer

VICE PRESIDENTS

Charles H. Benz Leonard E. Haas
Andrew A. Mickus Jerry M. Okabe
Donald A. Pazour Vicki L. Masseria

CHIEF FINANCIAL OFFICER

Daniel J. Mills

 **Miller Freeman**
A MEMBER OF THE UNITED NEWSPAPERS GROUP

600 Harrison St., San Francisco, CA 94107
(415) 905-2200 FAX (415) 905-2234



will be exploring these issues and kinds of products more. But one has to consider: As front-end tools pull further away from their DBMS roots—and as other, more foreign types of data access software press into the database—is the DBMS paradigm itself going to pull apart as it moves further from its roots?

Despite their wayward, data-centric leanings, at some point DBMSs have always had to make peace with their process-oriented parents, the operating system, and hardware. (And in a networked, client/server world, the DBMS will also have to acknowledge its eccentric uncles and aunts, network operating systems and protocols.) To quote Date again from *An Introduction to Database Systems*, "the DBMS has a view of the database as a collection of stored records, and that view is supported by the file manager; the file manager, in turn, has a view of the database as a collection of pages, and that view is supported by the disk manager; and the disk manager has a view of the disk 'as it really is.'" To solve performance problems, vendors (such as IBM, DEC, and other hardware companies) have typically integrated their DBMS products more tightly with the operating system and hardware. Such "bundling," of course, has been in part a marketing strategy to make it tougher for independent software vendors to loosen the hardware vendor's account control.

But even independent DBMS software vendors have headed their products in the direction of new hardware paradigms in pursuit of higher performance. Oracle, for example, has been making a lot of noise about a version of its DBMS that takes advantage of scalable, massively parallel supercomputers. This new version was made possible by an adjustment in v. 6.2, enabling the product to run on loosely coupled VAX clusters. Oracle has progressed past that point to the massively parallel machine, such as the nCUBE 2 Supercomputer, which reportedly eliminates bottlenecks by using a distributed memory structure in which each processing node consists of a processor and some local memory. Thus, multiple database servers

can run concurrently and independently, with their own memory for database buffers and backup and recovery facilities.

This development has enabled Oracle to exploit symmetric multi-processor environments offered by Sequent Computer Systems, Sun Microsystems, and several other hardware vendors. Interestingly, Oracle's direction is also pushing DEC a bit harder toward UNIX (that is, Ultrix). DEC VAX salespeople used to like Oracle (over its own DBMS, Rdb) because it helped them sell boxes; but now Oracle seems to favor other hardware, particularly systems running under UNIX. Rdb, a fine but scantily marketed DBMS (bundled with the VMS operating system, it's given away free), is rapidly being ported to UNIX. The blood feud between Oracle and DEC may be only just beginning.

One other famous solution to the database performance problem was the database machine, the first of which was marketed by Britton-Lee (now Sharebase and owned by Teradata Corp.). Bob Epstein, the founder of Sybase Inc. and a member of the University of California, Berkeley team that developed Ingres in the late 1970s, spent some time at Britton-Lee. He brought some of the database machine ideas with him to his own start-up, particularly the notion of an "intelligent" server that contains stored commands or procedures for later execution. "We want the database to model not only the data, but the business," Epstein said in a 1990 interview. Stored procedures and triggers have become fundamental concepts of today's client/server computing.

Sybase was also one of the first developers to build multitasking capability into the DBMS, independent of the operating system's functions. To describe this feature, I would like to quote from an excellent November 1991 Strategic Directive from New Science Associates, a consulting and research firm, written by John Girard (for more information on this report, call 203-259-1661): "Sybase and its competitors were conceived in the original single-processor UNIX and VMS environments. From the start, Sybase



How Borland's Paradox Is Making News in the USA . . . TODAY

USA TODAY® has changed the way Americans look at newspapers. As "The Nation's Newspaper," USA TODAY goes after critical, in-depth stories that affect people across the country.

But because many of those stories start with mountains of numbers and statistics, getting the facts is a difficult, time-consuming task. And in the deadline-oriented newspaper business, there is simply no time to waste.

To solve these complex problems, USA TODAY chose Borland's Paradox.®

The Dawn of "Database Journalism"

USA TODAY's Special Projects Unit has the daunting task of gathering and analyzing various government reports and statistics to get to the important stories hidden underneath.

They call their work "database journalism," and Paradox is the tool of the trade.

Reporters download information—from census figures to campaign finance data to crime statistics—into specially created Paradox databases for easy searching and analysis. In many cases, Paradox imports these figures directly, since it reads dBASE® and Lotus® files.

The S&L Scoop

In fact, Paradox and database journalism were key figures in USA TODAY's groundbreaking series of reports on the Savings & Loan crisis.

Early in 1989, USA TODAY's Special Projects

Unit used Paradox to investigate and analyze the health of more than 3,000 S&Ls across the country. As a result, USA TODAY readers were among the first to know about the national ramifications of this important story.

A Messenger Called AMANDDA

USA TODAY has 29 bureaus and regional offices nationwide. The paper's production depends on all those offices sharing messages and vital information.

To handle this information flow, a system called AMANDDA (for Automated Messaging AND Directory Assistance) was created using Paradox to work with the corporate electronic mail system. This remarkable partnership links every USA TODAY office, and provides instant access to a variety of information sources, the lifeblood of a daily newspaper.

And with Paradox, AMANDDA was up and running in less than two months.

Just the Facts

Paradox has all the powerful database features you need. Features like superior single- and multiuser access. Query By Example (QBE) to simplify finding the information you're looking for. Presentation-quality graphics for outstanding reports. The turbo-driven VROOMM™ system for maximizing memory use. Multi-table forms that let you look simultaneously at information from several tables. And that's only the beginning.

For developers, Paradox includes the comprehensive Paradox Application Language (PAL™). PAL puts the power of Paradox at your fingertips, and a set of robust development tools completes the programming environment. Exactly what you'd expect from a world-class database management system.

Some Independent Opinions

Some of the most respected industry publications have also made Paradox their #1 choice.

"With its combination of speed, ease-of-use, and practical features, Paradox [is] an excellent value."

—Computerworld

April 8, 1991

"Paradox is an outstanding product for developers and end users. It's one of the finest combinations of programming and interactive environments available."

—InfoWorld

November 12, 1990

Get Paradox. And Get the Whole Story.

Join USA TODAY and thousands of other companies who keep costs down and profits up with Paradox.

To order Paradox, see your dealer today or call 1-800-331-0877, Dept. 6264

Copyright © 1991 Borland International, Inc. All rights reserved. Paradox, VROOMM and PAL are trademarks of Borland International, Inc. BI 1445

B O R L A N D

Software Craftsmanship

CIRCLE 4 ON READER SERVICE CARD

chose to implement a more difficult but rewarding architecture for its access routines, based on 'threaded' processing. In mid-1985, most operating systems did not support threaded applications, so Sybase wrote its own thread manager. The impact was to allow Sybase to run with only one set of its access routines [termed an 'engine'] per processor, which served all connected users.

"Threaded code is similar in many respects to multitasking code, but offers some additional advan-

tages. First, threading allows the access engine to run as a single process within a single address space. This means that user applications communicate with the database engine through efficient shared memory paths rather than through more time-consuming interprocess communications mechanisms. Second, locks and contentions were also easier to manage, because they were controlled from a common, high-speed memory cache. Third, and most important, is that Sybase is in complete con-

trol of access routines and process behavior because it was not relying on the less-efficient and generalized control and I/O routines provided in each operating system... The Sybase architecture bypasses most of the general-purpose functions in the operating system, substituting its own, optimized, reduced-instruction routines. In fact, the underlying model is quite similar to that of a RISC processor design."

Epstein was a true pioneer of the client/server approach; Sybase recognized that in order to provide server intelligence and adequate response to OLTP demands, it had to gain control of more of the performance knobs and levers from the operating system. However, Sybase has not been immune to change. The company had to rework the product in 1990-91 so that its multithreaded architecture would work with symmetric multiprocessing systems.

Just as client/server computing offers front-end flexibility and independence, it is also shaking up the back end. Thus, for OLTP, some would suggest that we'll see the new class of transaction processing machines (such as Transarc's Encina) compete with database servers. Other start-ups, such as Red Brick Systems and UniSQL are developing specialized systems for better performance. I'm afraid whoever suggested that DBMS products were now commodities was far too premature.

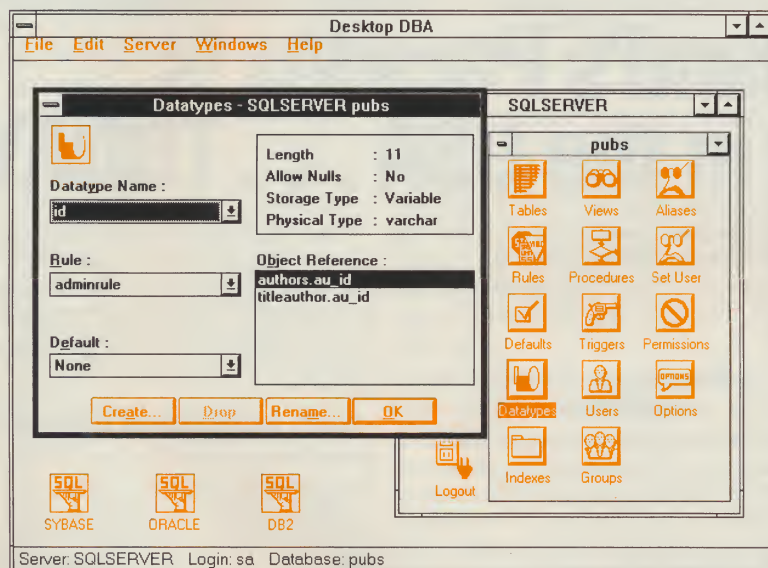
WE'RE OPENING ANOTHER communications avenue to our readers: a comment line. Thanks to the powers of voicemail, we have set up an extension for those who would like to leave us a brief message regarding articles and columns we've run, topics or products we should be covering, or even industry tidbits (rumors, hot leads, that kind of thing).

Now, the busy reader who hasn't the time to write can give an "instant" comment. The phone number is: (415) 905-2785. We look forward to your recording!

DAVID STODDER
EDITOR

Desktop DBA™

Serving up database servers... Windows style!



Desktop DBA delivers Windows® 3.0 front end power to match that of SQL database servers. With Desktop DBA, you can manage any number of database servers on your network simultaneously, each in its own window.

Desktop DBA goes beyond even the high end utilities available in mainframe environments, with features like drag-and-drop database migration. Point-and-click object management. Automatic corruption detection and repair.

All of this means you don't have to accept any more excuses about how client/server technology is lacking tools for DBAs and developers. With Desktop DBA, all the pieces are in place.

Desktop DBA for Microsoft® SQL Server™ and SYBASE® available now. Desktop DBA for Oracle® available early 1992.

DATURA The front end company

1926 East Parham Road
Richmond, VA 23228
804 264-1225
FAX 804 264-1297

CIRCLE 5 ON READER SERVICE CARD

MARCH 1992

SPECIAL OFFER!

SAVE
\$10

DEVELOP SOLUTIONS!

DATABASE PROGRAMMING & DESIGN: 12 Issues
for ~~\$47.00~~ — Now Only **\$37.00!**

YES! Please begin my subscription—12 monthly issues—for only \$37.00. I'll save \$10.00 off the regular subscription price of \$47.00. ☐ Payment enclosed ☐ Please bill me

Name

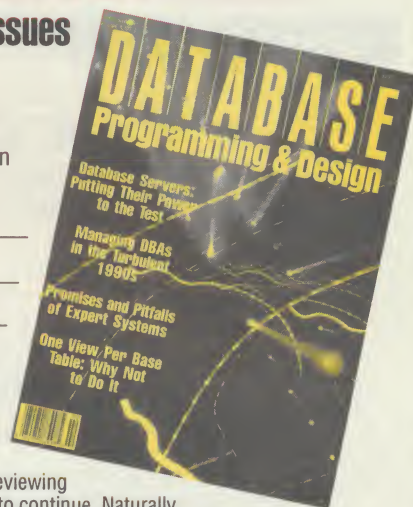
Company

Address

City State Zip

Please allow six weeks for delivery. CA state residents add \$2.68 sales tax. Foreign orders must be prepaid in U.S. funds. Canadian / Mexican orders add \$6 per year for surface mail. All other foreign add \$15 per year for surface mail or \$40 per year for air mail. Canadian GST #124513185.

PROFESSIONAL GUARANTEE: If I am not completely satisfied after reviewing my first issue, I may return your bill marked "cancel" if I do not wish to continue. Naturally, I may cancel my subscription at any time and still receive a refund for the subscriber price.



42CA

APPLY TECHNIQUES!

DATABASE PROGRAMMING & DESIGN: 12 Issues
for ~~\$47.00~~ — Now Only **\$37.00!**

YES! Please begin my subscription—12 monthly issues—for only \$37.00. I'll save \$10.00 off the regular subscription price of \$47.00. ☐ Payment enclosed ☐ Please bill me

Name

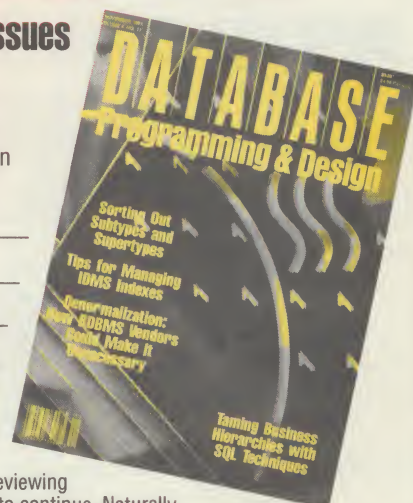
Company

Address

City State Zip

Please allow six weeks for delivery. CA state residents add \$2.68 sales tax. Foreign orders must be prepaid in U.S. funds. Canadian / Mexican orders add \$6 per year for surface mail. All other foreign add \$15 per year for surface mail or \$40 per year for air mail. Canadian GST #124513185.

PROFESSIONAL GUARANTEE: If I am not completely satisfied after reviewing my first issue, I may return your bill marked "cancel" if I do not wish to continue. Naturally, I may cancel my subscription at any time and still receive a refund for the subscriber price.



42CA

ESTABLISH CONTROL!

DATABASE PROGRAMMING & DESIGN: 12 Issues
for ~~\$47.00~~ — Now Only **\$37.00!**

YES! Please begin my subscription—12 monthly issues—for only \$37.00. I'll save \$10.00 off the regular subscription price of \$47.00. ☐ Payment enclosed ☐ Please bill me

Name

Company

Address

City State Zip

Please allow six weeks for delivery. CA state residents add \$2.68 sales tax. Foreign orders must be prepaid in U.S. funds. Canadian / Mexican orders add \$6 per year for surface mail. All other foreign add \$15 per year for surface mail or \$40 per year for air mail. Canadian GST #124513185.

PROFESSIONAL GUARANTEE: If I am not completely satisfied after reviewing my first issue, I may return your bill marked "cancel" if I do not wish to continue. Naturally, I may cancel my subscription at any time and still receive a refund for the subscriber price.



42CA

SAVE \$10

SUBSCRIBE TODAY!



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 1286 BOULDER, CO

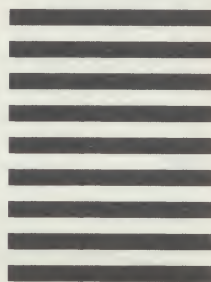
POSTAGE WILL BE PAID BY ADDRESSEE

DATABASE
Programming & Design

P.O. BOX 51247
Boulder, CO 80321-1247



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



SAVE \$10

SUBSCRIBE TODAY!



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 1286 BOULDER, CO

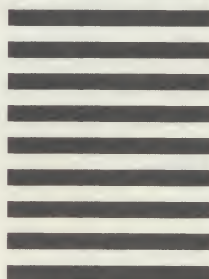
POSTAGE WILL BE PAID BY ADDRESSEE

DATABASE
Programming & Design

P.O. BOX 51247
Boulder, CO 80321-1247



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



SAVE \$10

SUBSCRIBE TODAY!



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 1286 BOULDER, CO

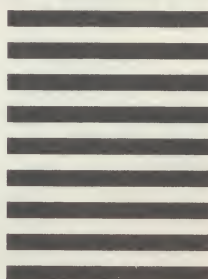
POSTAGE WILL BE PAID BY ADDRESSEE

DATABASE
Programming & Design

P.O. BOX 51247
Boulder, CO 80321-1247



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



**THE BEST
IN DATABASE
PROGRAMMING
AND DESIGN!**

The *only* magazine exclusively devoted to the professionals responsible for managing the corporate database—practical solutions for designing, using, and maintaining database management systems. Here's what you can look forward to *every month* in **Database Programming & Design**:

- Latest design techniques and products
- Proven design and programming solutions
- Tips for improved applications programming
- Ideas for trouble-shooting
- News and reviews on the latest hardware and software
- Case histories

Time and money-saving guidance in database development, construction, and maintenance at a *special* subscriber price!

To start your subscription, detach and mail one of the post-paid cards today! You'll get one year of **Database Programming & Design**—*12 monthly issues—at the special rate of \$37.00.*

A savings of \$10.00!

SPECIAL SUBSCRIPTION OFFER!

SAVE \$10

DATABASE
Programming & Design

CICS
printf("Hello world!");

**Now the SAS/C® compiler is
the perfect choice for
CICS applications, too.**

Now you can develop command level CICS applications with the industry's leading main-frame C compiler, the SAS/C® compiler from SAS Institute. Running under MVS or VM/CMS, the SAS/C compiler and CICS preprocessor can be used to develop transaction-oriented applications for CICS (Release 1.7 or above) in MVS, MVS/ESA, and DOS/VSE environments.

With the SAS/C compiler's command preprocessor, all CICS commands, including DL/I and BMS, can be freely interspersed with C code. And that includes support for commands that other C compilers simply can't handle (such as HANDLE CONDITION, IGNORE CONDITION, PUSH/POP HANDLE and HANDLE AID).

File I/O for CICS transient data queues (both extrapartition and intrapartition) and JES spool files are fully supported through standard C library functions. Run-time storage analysis and usage reports are also available. And with our exclusive support for JES spool files, you can retrieve a file from the JES spool, write a file directly to the JES spool, and send a file to a remote destination via systems connected to a JES/RSCS network.

As you would expect, the SAS/C library functions for memory allocation, math and string handling, diagnostic control, inline machine code, and dynamic loading also work under CICS. And for error checking, the SAS/C library provides signal handling, full tracebacks, and a complete set of warning/error messages.

The SAS/C transient library is freely redistributable with your CICS applications. There are no hidden run-time fees. Or you can use our exclusive All-Resident Library to automatically link the necessary run-time routines directly into your application and eliminate the need for a transient library altogether.

 SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Phone 919-677-8200
Fax 919-677-8123

CICS Systems Programming Too!

The SAS/C compiler set the standard for systems programming in C. And now our exclusive Systems Programming Environment makes easy work of writing CICS exits and utilities as well.

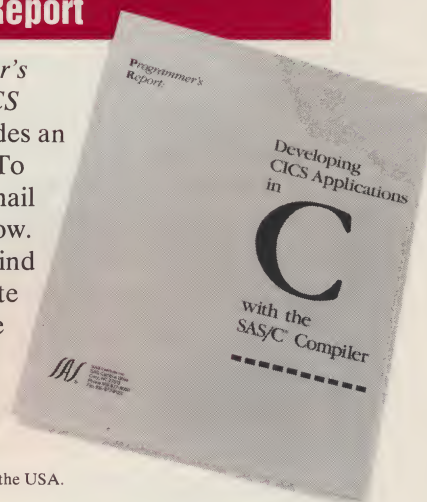
With frequent updates and knowledgeable technical support—both provided free—the SAS/C compiler is the best investment you can make toward more effective and maintainable CICS applications.

**Learn More With a Free
Programmer's Report**

Our free *Programmer's Report: Developing CICS Applications in C* provides an excellent introduction. To receive a copy, simply mail or FAX the coupon below. Or call us right now to find out how you can evaluate the SAS/C compiler free for 30 days.

SAS and SAS/C are registered trademarks of SAS Institute Inc., Cary, NC, USA.

© 1991 by SAS Institute Inc. Printed in the USA.



☐ Please send me your free *Programmer's Report: Developing CICS Applications in C*.

☐ Please send me details of a FREE 30-day trial of the SAS/C Compiler.

Please attach your business card, or fill out below:

Name

Title

Company

Address

City State ZIP

Phone

Mail to: SAS Institute Inc.
Attn: ME, SAS Campus Drive
Cary, NC, USA, 27513
Phone 919-677-8200 FAX 919-677-8123

DB3/92

Our data dictionary organizes even better than she does.

With IBM's AD/Cycle on the horizon, getting your meta data in order is long overdue. Fortunately, DB EXCEL[®], the innovative family of repository-based tools to support the IBM SAA development and maintenance environment, makes it easier than ever to collect and maintain information about your enterprise systems.

DB EXCEL is a DB2-based data dictionary with a long list of unique features that makes it the ideal way to prepare for a successful migration to IBM's Repository and AD/Cycle. DB EXCEL supports both DB2 and IMS. It offers bidirectional interfaces to popular CASE tools like Bachman, KnowledgeWare's IEW/ADW and Intersolv's Excelerator, for leveraging your investment in those technologies.



Everything about DB EXCEL is designed to make standards implementation easy. Customization is simple, thanks to DB EXCEL's automated extensibility. And the PC-like pop-up windows get users defining and maintaining data entities quickly. Even repository editing is a breeze, with a powerful edit facility that supports editing multiple entities on one screen (mass edit) or sequential editing of a list of entities (queue edit).

As an IBM Business Partner, we made sure that DB EXCEL's flexible, dynamic architecture fits in with IBM's plans for AD/Cycle. DB EXCEL

can allow smooth implementation of RM/MVS in the future, and can function as an integral part of that solution once it has been installed, so your investment in DB EXCEL is protected.

To make using DB EXCEL even simpler, we back it with superior service, from hands-on installation assistance to

DB EXCEL[®] The innovative family of repository-based tools

detailed, quality manuals and documentation.

Check out DB EXCEL for yourself—with our 30-day free trial.

Call 1-800-333-4899 to arrange for an on-site demo.



Suite 450 • 3211 Jermantown Road
Fairfax, Virginia 22030-2844

DB EXCEL is a registered trademark of Reltech Products. All other trademarks are proprietary to their respective manufacturers.



CIRCLE 6 ON READER SERVICE CARD

ACCESS PATH

Lauds and lectures, kudos and critiques

DEAR EDITOR:

I would like to respond to Barbara von Halle's April 1991 column ("The Year in Review," Database Design) in which she says, "The migration from application-oriented (narrow-scoped) databases to shared databases is as much a cultural change as a technological one." Actually, I think the change is more cultural than technological. Most systems-development departments and senior-level managers are rooted in the '70s and believe that data sharing attacks the culture that has built up over the last 20 years. Any data administration manager trying to foster a shared-data environment should keep a resumé handy and remember Machiavelli: "It must be considered that there is nothing more difficult to carry out, nor more doubtful of success, nor more dangerous to handle, than to initiate a new order of things" (*The Prince*, Oxford University Press, 1979).

I recently attempted to educate my boss and his colleagues on the shared-data environment's importance and steps for its implementation. However, under the guise of a "relative importance factor," my position as manager of data administration was eliminated and I was "let go." Of course, the culture was saved.

Has any other company had success implementing data sharing?

Keith C. Hulslander
Dover, Pennsylvania

Columnist Barbara von Halle responds: Hulslander's situation is a sad one and I tried to solicit some immediate answers to his letter.

Regarding data sharing, Deborah L. Brooks, data specialist at a Washington D.C.-based corporation, said, "The cultural change toward data sharing is slow but sure. With increased pressure from domestic and

We would like to hear from you! If you have suggestions or comments about our publication, write: Editor, DATABASE PROGRAMMING & DESIGN, 600 Harrison St., San Francisco, CA 94107.

overseas competitors, American corporations are forced to define their corporate strategies clearly. With this shift in focus to corporate-level resources, it's beginning to be recognized that data transcends organizational boundaries." Brooks believes the days of individual fiefdoms with closely guarded information stores are numbered.

Carolyn Kohler, manager of data design coordination at Levi Strauss & Co. in San Francisco, said that data sharing has become a reality in her company, and agreed that the change is more cultural than technological. She explained, "Levi Strauss & Co. embarked on the path of data integration a few years ago and has succeeded in building an enterprise model and involving the users in the gathering of data requirements and implementation of five shared subject area databases."

Kohler believes that Levi Strauss's data sharing success was due to the support and commitment from Information Resources management (from the CIO down) as well as the user community, and a companywide "I'll try it" attitude.

DEAR EDITOR:

Database Programming & Design is an excellent, professional product that achieves industrywide recognition. By far its weakest attribute, however, is the Celko on SQL column. Celko presents a naïve, flip-pant attitude and viewpoint on most issues and isn't consistent with the quality of your other authors. In particular, I'd like to respond to Celko's views on down-

sizing ("On Top of the World," Celko on SQL, August 1991).

Celko states that "now downsizing means replacing large mainframe computers with networked PCs." Although I don't disagree that downsizing is a trend in the industry, defining downsizing as a replacement of large mainframes with networked PCs is inaccurate. As Celko suggests, some applications could downsize to networked PCs, but not the majority of applications running on large mainframe computers. In addition, his column suggests that downsizing to PCs is easy and inexpensive. This attitude is naïve and requires further elaboration.

Although I'm not suggesting that I know the true definition of downsizing, I would like to distinguish between a downsized environment and a large mainframe environment. A downsized environment would most likely have the following characteristics:

- ☐ A greater number of independent computers;
- ☐ Larger physical distribution of computers;
- ☐ More sophisticated network and application software, system/configuration management, and professionals;
- ☐ Higher initial as well as sustaining costs;
- ☐ Greater user productivity (not cost savings).

The large mainframe computer environments migrating to a downsized environment will likely view the mainframe as a data and information server for the enterprise. Therefore, some portion of the mainframe complex will remain part of the overall computer infrastructure.

A properly selected migration path from large mainframe computers should result in a more productive environment for users,

but probably won't result in less actual computing costs. As the preceding list implies, more costs are associated with downsizing than cost-avoidance with cheaper MIPS on the desktop. As with any decision, the business case drives the solution.

Michael Krasowski
Mission Viejo, California

Columnist Joe Celko responds: I have over 200 publication credits in the computer trade press, so I think I have my column down pretty well by now.

In addition, I've managed to earn a decent living as a consultant, which isn't exactly a job for the naïve.

Krasowski's remark that the majority of programs on mainframes can't be moved down to smaller machines is interesting. Can he prove it? I'll give a counter example. Dr. George Schussel of Digital Consulting Inc. has a client who moved a 130,000-line COBOL program from an IBM mainframe to Micro Focus's COBOL. This conversion required only 12 changes in the source code and now runs faster.

As far as the mainframe acting

as a file server, I have the same doubts as Dr. Schussel. From my own experience, I can give Krasowski details on large- to medium-sized Canadian banks that work on networked PCs without a single mainframe.

I wonder why Krasowski assumes that a downsized environment will have a greater number of independent computers and larger physical distribution. I look at traditional mainframe shops today and every PC is used as a word processor and spreadsheet machine. Does this mean a shop is downsizing if everyone owns a pocket calculator? I don't think so. The defining characteristic is access to corporate data resources, which implies some type of network.

I'm not sure what Krasowski means by "sophisticated," but I'll assume it means complex or expensive. Indeed, I hope we'll see more sophisticated network systems, where systems do all of the hard work so that we don't have to hire sophisticated professionals. I have high hopes that the deal between Digital Research and Novell will produce a high-quality, integrated network operating system that blows the socks off those we have now, which are weak on recovery, access, logging, and security. Traditionally, these systems were tacked on top of existing single-user operating systems, not integrated into them.

I expect to see more application packages on smaller machines as time goes on, which will probably be ported down from mainframes. When I can get the same commercial package on a PC as a mainframe, I'll know the mainframe has died completely.

A pattern seems to exist in computing that makes life easier for the user. When mainframes were first developed, users had to write custom software for every application. When minicomputers came out, the same thing occurred. Why should networked PCs be any different?

I don't understand Krasowski's idea that greater productivity won't result in cost reduction. If we assume that productivity is measured in transactions per day, the only way that a greater number of transactions per day could result with the same cost per transaction would be to increase other costs. And does Krasowski think that maintenance costs for PCs is higher than for mainframes? I wonder how many PCs I can buy and service for the monthly rental on one mainframe?

DB2

PUBLIC SCHEDULE

* DB2 Data Base Design & Administration (4 days) \$825

New Haven, CT.....	Apr	6-9
Orlando, FL.....	Apr	14-17
Lincoln, NE.....	May.....	11-14
Boston, MA.....	May.....	18-21
Long Beach, CA.....	May.....	26-29
San Francisco, CA.....	Jun.....	9-12
Dallas, TX.....	Jun.....	23-26
Philadelphia, PA.....	Jul.....	21-24
Chicago, IL.....	Jul.....	28-31
Stamford, CT.....	Aug.....	24-27
Miami, FL.....	Sep.....	22-25
New Haven, CT.....	Oct.....	19-22
Atlanta, GA.....	Oct.....	26-29
Long Beach, CA.....	Nov.....	3-6
Boston, MA.....	Nov.....	16-19

* Advanced SQL (2 days) \$575

Boston, MA.....	Feb.....	20-21
Hartford, CT.....	Mar.....	5-6
Orlando, FL.....	Mar.....	19-20
Columbus, OH.....	Apr.....	21-22
Washington, DC.....	May.....	7-8
New Haven, CT.....	May.....	20-21
Hartford, CT.....	Jun.....	4-5
Lincoln, NE.....	Jun.....	15-16
Long Beach, CA.....	Jul.....	9-10
Boston, MA.....	Jul.....	16-17
Philadelphia, PA.....	Aug.....	20-21
San Francisco, CA.....	Sep.....	17-18
Dallas, TX.....	Sep.....	24-25
Boston, MA.....	Oct.....	1-2
Atlanta, GA.....	Oct.....	22-23
New Haven, CT.....	Nov.....	5-6
Miami, FL.....	Nov.....	19-20

Other classes available: ADW, CICS, CSP, OS/2, IMS, C, SQL/DS, REXX, TSO, VM, COBOL, JCL and Dialog Manager

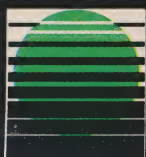
* DB2 Application Programming (4 days) \$825

New Haven, CT.....	Feb.....	24-27
Boston, MA.....	Mar.....	9-12
Hartford, CT.....	Mar.....	16-19
Lincoln, NE.....	Apr.....	13-16
Washington, DC.....	Apr.....	20-23
Long Beach, CA.....	Apr.....	21-24
New Haven, CT.....	May.....	4-7
Dallas, TX.....	May.....	11-14
Boston, MA.....	Jun.....	8-11
Philadelphia, PA.....	Jun.....	22-25
San Francisco, CA.....	Jul.....	20-23
New Haven, CT.....	Aug.....	17-20
Chicago, IL.....	Aug.....	24-27
Hartford, CT.....	Sep.....	14-17
Long Beach, CA.....	Oct.....	6-9
Atlanta, GA.....	Oct.....	12-15
Miami, FL.....	Nov.....	2-5
New Haven, CT.....	Nov.....	9-12
Boston, MA.....	Dec.....	7-10

* QMF (Query Management Facility) (2 days) \$575

New Haven, CT.....	Mar.....	9-10
Chicago, IL.....	Apr.....	16-17
Boston, MA.....	Jun.....	25-26
New Haven, CT.....	Sep.....	10-11
Boston, MA.....	Oct.....	29-30
Chicago, IL.....	Nov.....	19-20

* All classes feature machine workshops



BENCHMARK TECHNICAL SERVICES

386 Main Street, Middletown, CT 06457
(203) 346-4661

CHARTER
VENDOR
MEMBER



INTERNATIONAL
DB2 USERS GROUP

CIRCLE 7 ON READER SERVICE CARD

MARCH 1992

DB2 buffer pools: sink or swim



You can drown in wasted buffer pool space or run aground if you don't adjust your buffer pool sizes based on workload needs. But you can't afford the downtime associated with changing buffer pool sizes, so you do what you can to stay afloat.

Announcing **OPERTUNE™**

Now for the first time, you can achieve continuous availability while changing buffer pool sizes. This capability is available only with OPERTUNE™ for DB2* from BMC Software.

When peak processing times require larger buffer pools, you can increase the sizes to meet your needs, reducing bottlenecks and speeding response times. Likewise, when requirements call for reduced buffer pools, you can quickly and easily modify them, saving storage space.

OPERTUNE lets you achieve optimum performance while maintaining continuous data availability by allowing you to dynamically tune for your changing DB2 needs.

Dynamically changing buffer pool sizes is achieved with BMC Software's exclusive technology. It uses DB2 to implement many changes and requires absolutely no USERMODs or dynamic hooks.

Take control

Achieve continuous availability while tuning for optimum performance. For more information about changing DB2 buffer pool requirements or any of the additional OPERTUNE features, contact BMC Software today at 713 240-8800 or **1 800 841-2031**.

BMC SOFTWARE

The Experience. The Technology. The Future.

BMC Software international offices are located in Australia, Canada, Denmark, France, Germany, Italy, Japan, Spain and the United Kingdom.
*DB2 is a registered trademark of International Business Machines Corp. ©1992, BMC Software, Inc. All rights reserved.

CIRCLE 8 ON READER SERVICE CARD





Taming DB2 with TABLES

Improve application development productivity 20-50 times.

Prototype and maintain DB2 tables in production CICS and IMS without programming... distribute production table maintenance to end users... bring your table applications on-line by as much as 50 times faster than using CASE tools... And do this with the full confidence of SAA compliance.*

Maintenance screens can be created by simply naming a table. Editing, validation, and processing logic, if needed, can

The Power To Make It Simple

- Applications up to 50 times faster than CASE tools
- Production on-line maintenance
- Static SQL
- Common memory dataspace five times faster than DB2 buffers
- Time dependent row processing
- Menus and help screens

be added using TABLES custom development facilities.

For ultra high performance, (5x faster than DB2 buffers) tables can be loaded into common memory dataspaces and concurrently shared across multiple regions including: CICS, IMS, TSO and batch.

For a free trial or to learn more about how TABLES is bringing new meaning to the term productivity,

call (800) 328-2825

 **SPECIALIZED
SOFTWARE**
INTERNATIONAL
THE TABLE MANAGEMENT COMPANY™

* Registered in IBM's SAA Catalog

DB2 is a trademark of the IBM Corporation

DATABASE DESIGN

The data-sharing lesson is tough to learn, but it's a worthwhile exercise

HAVE YOU EVER TRIED to teach a child the value of sharing? The first lesson usually proceeds along these lines: "You should share your possessions with your siblings, friends, and even people who aren't your friends because it's a nice thing to do—it makes the other person feel good." Then, it progresses to: "No, I won't buy another one—you can both use the same one . . . Why? Because I said so!" And finally: "If you don't share it, I'll take it away, and no one will have it!" With lessons like this one, no wonder our perceptions of sharing are somewhat less than optimal.

My six year-old son has developed an interesting strategy for sharing his Teenage Mutant Ninja Turtles with his four year-old brother. He bases it on four carefully constructed "sharing principles." First, he and his brother never play with the same turtle at the same time. Second, his brother "shares" only one turtle at a time. Third, his brother can "share" a turtle only when the older one isn't playing (or even thinking of playing) with it. Fourth, and most importantly, the turtles aren't community property that can be leveraged for the good of both siblings. In fact, the younger sibling sacrifices all sharing rights the moment he refers to them as "our turtles."

Obviously, most of us never want to share anything. But it's interesting that a lot of us have chosen a career in which "wide-scale sharing" is the ultimate goal. However, what should we do when familiar tactics, such as removing the "shareable object" or refusing to duplicate it, don't work in the corporate world?

For starters, we have to find more civilized data-sharing strategies. Two months ago, we began

BY BARBARA VON HALLE

Share and Share Alike

developing such a strategy using John Zachman's Information Systems Architecture Framework (ISA Framework). The framework aims at determining an architecture for a single information system. We extended this scope to include our entire data-sharing universe, which, by definition, encompasses many information systems.

Since last month, we've been suspended in the second row of the data column labeled Business Data Model. We arrived here by solidifying the data-sharing scope, organizing the business community around this scope, and determining the specific purpose for developing the business data model. We also accepted Zachman's suggestion that this cell results in business entities and relationships. However, alternatives such as object and Nijssen's Information Analysis Methodology (NIAM) models are possible as well.

While poised in this cell, we questioned what its deliverables look like. We proposed two kinds of deliverables: a conceptual, planning-level business data model and a detailed, working-level business data model. This cell probably wasn't meant to house these two data models. Rather, Zachman and others place the detailed business data model in the next row of the framework labeled Information Systems Model.

This month, we'll explore the

conceptual and detailed business data models, and discuss where each may fit in the ISA Framework and our evolving data-sharing strategy.

A CONCEPTUAL business data model depicts high-level entities; however, it sometimes represents data groups or data areas rather than normalized entities. A conceptual data model for a large enterprise may consist of 10 to 30 entities, such as *product*, *customer*, *external organization*, *order*, and so on.

The conceptual business data model may also contain many-to-many relationships without resolving them into one-to-many relationships. In addition, it may include the most important subtypes without identifying all subtypes. And in some cases, relationships may not have names.

To validate the conceptual data model, we present it or some of its aspects to the business community. Therefore, we need to determine how well businesspeople will understand the data model. In some enterprises, higher business authorities may never view the data model diagram, and may only focus on the business definitions underlying them. In these instances, they approve business definitions for entities, business owners for these entities, and identifiers for entities that don't already have appropriate identifiers. The emphasis is on ensuring the most comfortable level of supertypes and subtypes possible.

In other enterprises, higher business authorities participate in a structured walk-through of the data model diagram. Their members can perform this walk-through as a group or individually. Individually, the reviewer can concen-

trate on those areas "owned" or "accessed" by the business area of immediate interest.

As we saw last month, the next step in our data-sharing strategy is to develop a detailed business data model. This model should be a fully attributed, normalized, keyed, and subtyped data model containing essential business information. A better name for this model is a detailed (not conceptual), business-oriented (not information systems-oriented) data (not process) model. In fact, a particular detailed data model may represent a smaller scope than the conceptual business data model.

A lot of controversy exists over where this kind of data model fits into the ISA Framework. Even speakers at the Zachman Framework for Information Systems Architecture Conference (Arlington, Virginia, March 25 to 27, 1991) didn't agree on whether attributes and keys first appear on the business model, information systems model, or technology model rows. Questions arose as to which properties of entities, relationships, and attributes are business-driven and which are information systems-driven.

To answer these questions, we should determine the audience. How much of this model does the business community need to be aware of, and how much of it concerns only information systems professionals? Because we build this model according to concepts that originated with the relational model, some may insist that it's an information systems data model. However, remember the important role of the business community in data sharing. The detailed, business-oriented data model represents, in an unambiguous way, the information essential to running a business, not a system. An information systems data model, on the other hand, should include the data needed to build a computerized system, including codes, processing flags, and other information systems or applications-specific data structures.

Regardless of where it fits in the framework, we must still develop and gain approval for a detailed, business-oriented data model. It isn't likely that the higher

business authority will review a detailed data model or detailed business rules; rather, approval for the detailed business data model will probably involve intermediate business management and day-to-day model participants. The business data model review sessions may also include business analysts and database designers.

With this understanding in mind, let's investigate the first of many controversial questions surrounding the content of a detailed business-oriented data model. As always, I welcome reader input.

Question one: Should entities in the detailed, business-oriented data model be fully normalized? The detailed, business-oriented model should represent the essential business information in a way that encourages data sharing. Therefore, it requires discipline—such as normalization, good primary identifiers, and subtype and supertyping. In fact, a detailed, business-oriented data model includes resolution of many-to-many relationships (not to imply a particular information systems perspective, but to uncover more detailed business rules about the data). Often, it's useful to present such details to businesspeople.

Question two: Should arbitrary, artificial entity identifiers be included in the detailed business-oriented data model? Most of us will agree on one surprising fact: The most important entities in an enterprise's business don't have definitions that are commonly understood by everyone in the business community. It's even more likely that the most important business entities don't have adequate identifiers. As Theresa Whitener suggested in her article entitled "Primary Identifiers: The Basics of Database Stability" (*Database Programming & Design*, January 1989), a good identifier is:

- ☐ Unique for each object instance;
- ☐ A definite value for every object instance;
- ☐ Explicit when an object instance is created;
- ☐ Only one consistent value for every object instance;
- ☐ Unchangeable;
- ☐ Not fact-giving data;
- ☐ Legal to implement;

☐ Fully controllable by the enterprise;

☐ Visible to and accessible by users.

Most of us will agree that a good solution (and current industry trend) is the introduction of new, arbitrary, and nonintelligent identifiers. The question is whether these "imaginary keys" should be part of the business-oriented data model. Many people believe that these keys are "database keys" and therefore shouldn't be used for business consumption. However, I agree with Whitener that such identifiers should become visible to the business community whenever possible because the business needs them to share data effectively through a common, stable identification scheme. Therefore, we should include the identifiers in the business-oriented data model.

Because people react negatively to such identifiers, we can use the higher business authority to introduce them in a politically palatable way. We should educate the higher business authority in the properties of good identifiers as well as the present inadequacies of current identifiers. We should also gain its approval for any new identifiers. In fact, as we discussed earlier, we can include a few of these identifiers in the conceptual data model.

Question three: Should decode entities be included in the detailed, business-oriented data model? Decode entities are "look-up" entities that transcribe an abbreviated code into its full business meaning. An example is an entity called *employee type* that translates EMPLOYEE TYPE CODE (*p* or *f*) into employee types, part- and full-time.

The existence of decode entities in business data models can result in a lot of emotional discussion. From a purely business perspective, such an entity is superfluous to the business community, even though the community is accustomed to the codes. In reality, the code isn't essential to running the business. The business community merely needs to distinguish between the part- and full-time employees.

I've met at least two people who include decode entities in the

Logic Works ERwin™/ERX

THE PRODUCT EVERYONE IS TALKING ABOUT.

Here's What the Experts Say:

"Maybe, someday, all databases will be designed this way."

—DBMS, November 1990

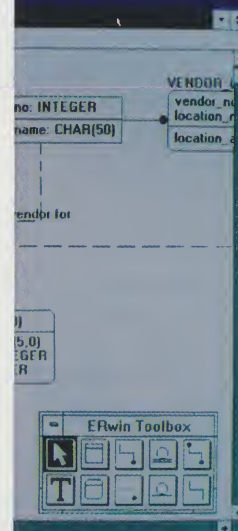
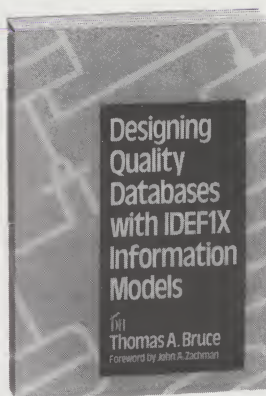
Here's What We Say:

Tom Bruce is talking about ERwin!

Why has ERwin become the design tool of choice? Tom Bruce's new book, *Designing Quality Databases*, explains the IDEF1X method, with real-world examples featuring ERwin as the design tool that makes it happen.

We think Tom Bruce's book will help you get even more from ERwin tools. So, for a limited time, we're including it FREE with every ERwin/ERX purchase.

☐ Tell me more about ERwin and *Designing Quality Databases with IDEF1X Information Models*.



Name _____
Title _____
Company _____
Address _____
City _____ State _____ ZIP _____
Phone _____

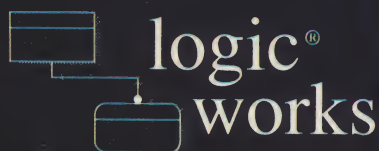
For faster response, call Logic Works today, at (609) 683-0054 or FAX (609) 924-0029.

Network SQL

- ORACLE
- SQLBase
- SQL Server and
- SYBASE



See us at Booth #616
Moscone Convention Center
San Francisco, CA
March 24 - 26, 1992



Logic Works
601 Ewing Street, Suite B7
Princeton, New Jersey 08540
(609) 683-0054 • Fax (609) 924-0029

ERwin Photo Credits: Michael Carr, Cynthia Engler. ERwin Diagram: Robert G. Brown.
Logic Works is a registered trademark and ERwin is a trademark of Logic Works.

CIRCLE 10 ON READER SERVICE CARD

Logic Works ERwin™/ERX

THE PRODUCT EVERYONE IS TALKING ABOUT.

Here's What the Experts Say:

"Maybe, someday, all databases will be designed this way."

—DBMS, November 1990

Here's What We Say:

The experts are right. Everyday more and more Big 8* databases are being designed with ERwin's logical data models and SQL schema generation.

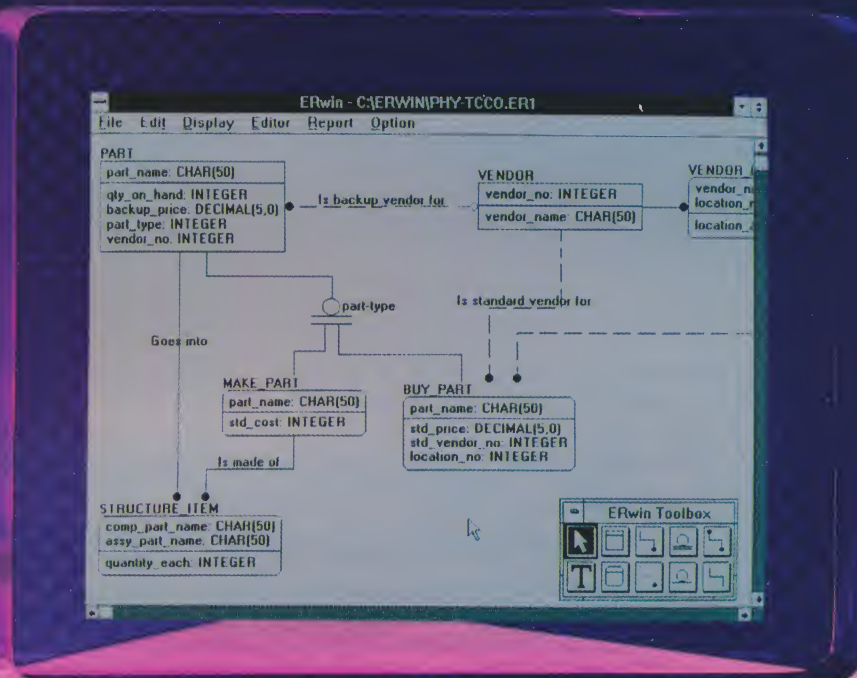
But why wait for "someday" when you can start designing your Big 8 database applications the right way today? What everyone is saying is true.

No other CASE tool makes capturing business rules, generating SQL DDL, and reverse-engineering your existing Big 8 databases into ER diagrams this easy.

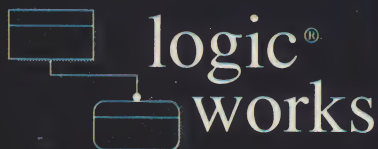
Call Logic Works today! For a limited time, ERwin/ERX, the product everyone is talking about, is available for only \$1995.

ERwin/ERX supports the Big 8*:

- DB2
- Informix
- Ingres
- Netware SQL
- ORACLE
- SQLBase
- SQL Server and
- SYBASE



See us at Booth #616
Moscone Convention Center
San Francisco, CA
March 24 - 26, 1992



Logic Works
601 Ewing Street, Suite B7
Princeton, New Jersey 08540
(609) 683-0054 • Fax (609) 924-0029

ERwin Photo Credits: Michael Carr, Cynthia Engler. ERwin Diagram: Robert G. Brown.
Logic Works is a registered trademark and ERwin is a trademark of Logic Works.

CIRCLE 10 ON READER SERVICE CARD

WHAT HAPPENED AT LAST WEEK'S ORACLE USERS GROUP MEETING?

Users groups are becoming an increasingly prominent vehicle for the flow of technical tips and information about DBMSs, CASE, and related software.

Therefore, we are looking for members of users groups to write about their groups' experiences for *Database Programming & Design's* Birds of a Feather column.

The column covers a variety of issues, from a technical and nontechnical standpoint. Future issues will include such topics as:

- Profiles of particular users groups: industries represented, levels of experience
- Hot ideas, problems, possibilities, and technical tips that come up during meetings
- Organizational issues and problem solving
- How to get the best speakers
- Approaching vendors from a nonmarketing standpoint
- Calendars of events
- Reports of recent meetings
- Making recommendations to and getting feedback from vendors
- How to use user groups as vehicles for change

If you would like to share your experiences with your users group—no matter how big or how small—we want to hear from you!

Theresa Rigney
Database Programming & Design
600 Harrison St.
San Francisco, CA 94107
(415) 905-2482

information systems or technology data model, rather than the business-oriented model. This philosophy is especially true when the decision to use codes is made to save space on DASD or screens. You may include these codes in computerized systems to satisfy business demand for them. Even so, excluding them from the business-oriented data model reinforces that they're unessential.

OUR ROLE AS INFORMATION specialists is an opportunistic one. We alone may have the vision, immediate need, and methods for integrating the enterprise. If we do our jobs well, we'll be the enablers of common organizational semantics and business policies.

We can use the ISA Framework as a guide for developing a data-sharing strategy. We have begun such a strategy over the past few months. It probably doesn't matter whether we place the detailed business data model in the second or third row of the framework, but we must make a distinction between a business-oriented data model and an information-systems-oriented data model. We should use the former to understand the business, and the latter to design databases for information systems. If, for whatever reason, we combine aspects of both, we should at least indicate which aspects are business-driven.

Regardless of our strategy, we should always be aware that sharing isn't necessarily universal in its definition. We must lead the business community carefully through the lesson of enterprise-wide sharing. It's a tough lesson, however, because multiple owners must often share a common asset. ■■■

REFERENCES

Zachman, J. "A Framework for Information Systems Architecture," *IBM Systems Journal*, 26(3): 276-292, 1987.

Barbara von Halle is director of technology planning for Spectrum Technology Group Inc., a North Branch, New Jersey-based consulting company. She is coauthor of *The Handbook of Relational Database Design* (Reading, Massachusetts: Addison-Wesley Publishing Co., Inc., 1989).



INFOSCOPE polishes DB2 quality and performance —automatically

At last there's a way to guarantee the quality, security and performance of DB2 applications. You can do it by using InfoTel's INFOSCOPE for DB2 to limit deviations from your established programming standards.

INFOSCOPE for DB2 ensures quality by enforcing preestablished programming standards via an ISPF interface.

INFOSCOPE for DB2 ensures security by barring the use of statements that are dangerous or prohibited under your standards.

INFOSCOPE for DB2 ensures performance of new or existing DB2 applications by demanding well-written SQL statements, efficient tablespace scans,

systematic memory strategies and straightforward access paths that minimize contention for resources.

INFOSCOPE for DB2 actually provides a mechanism for measuring quality by assigning costs to access strategies and language elements. Element costs are accumulated, and INFOSCOPE for DB2 issues approvals, warnings, or outright rejections. These are based on total cost, relative to the rules **you have established.**

Perhaps the most important feature of INFOSCOPE for DB2 is its flexibility. You can adopt and supplement quality rules to fit your particular computing environment; then you may modify them as your applica-

tion evolves and your environment changes.

Find out how you can write gleaming new DB2 applications and polish up your existing ones. Get full details and schedule a demonstration of INFOSCOPE for DB2 by contacting InfoTel today!

InfoTel

InfoTel Corporation
15438 N. Florida Avenue, Suite 204
Tampa, Florida 33613
1-800-543-1982 • 813-264-2090
FAX: 813-960-5345

CIRCLE 11 ON READER SERVICE CARD

DBAs... How Many Tools Does It Take To Manage DB2®?

As a DBA, managing an IBM® DB2 system falls on your shoulders. Daily, you're faced with a host of manual, time-consuming and error-prone procedures as you attempt to implement and maintain your DB2 system and support your DB2 programmers.

To manage your DB2 environment, you may be relying on a combination of the different DB2 tools and utilities currently available. However, this approach only adds another layer of complexity to an already complicated situation. You're faced with incompatible tools for exchanging information; learning new command sets for each tool; and, finally, working with tools that are generally difficult to use.

Now there is one product that provides the wide range of capabilities you need to meet all of your DB2 administration requirements — **TransRELATE Workbench for DB2** from Compuware.



One Source For All DBA Administrative Needs

TransRELATE Workbench for DB2 consolidates these essential capabilities:

- Comprehensive catalog query and analysis — without coding SQL
- Extensive DB2 object creation and management
- Extended object modification
- Automated migration
- Dropped object recovery
- Automated DB2 utility generation
- Simplified DB2 authorization management
- Backup and recovery JCL generation
- Automatic DCLGEN generation
- Fully-functional DB2 command interface — enter DB2 commands from within the Workbench

One Powerful Interface

The Workbench features a window-driven user interface that gives you easy and rapid access to vital DB2 data. Multiple views of one or more catalogs can be

simultaneously displayed on a single screen, helping you quickly analyze and solve problems.

Unique Maintenance Migration Capabilities

Exclusive Workbench capabilities include "intelligent" maintenance migrations. **TransRELATE Workbench for DB2** automatically identifies the impact of merging changes to existing DB2 subsystems — reporting all processing steps required — and automatically generates migration procedures.

One Product, a World of DB2 Benefits

By simplifying the complex process of environment management, the **TransRELATE Workbench for DB2...**

- Increases DBA productivity
- Reduces errors
- Shortens time to production
- Ensures reliable production support

Compuware is the leading provider of Computer Assisted Testing and Implementation (CATI) tools, designed to help IS professionals efficiently test and maintain applications. More than 60% of IBM MVS and ESA mainframe installations worldwide use Compuware CATI products.

For more information, or a no-cost 30-day evaluation, call

1-800-535-8707



COMPUWARE
Sophisticated Software Used Every Day

IBM and DB2 are registered trademarks of International Business Machines Corporation. Compuware and TransRELATE are registered trademarks of Compuware Corporation.

CIRCLE 12 ON READER SERVICE CARD

Domain classification: a scalar approach to consistent data names

BY ROBERT TAKOUSHIAN

The Naming Game

THE NEXT TIME YOU tell someone you're in the information management, communications, or data processing business, don't forget to mention that you're also in the transportation business. This statement is less of an exaggeration than might you think: As a data administrator or database administrator (DBA), your responsibilities are likely to include the movement and storage of critical company assets. And just as a shipping document must identify each item in a particular shipment, the data administrator or DBA must also be able to identify and name whatever data is moved or stored.

Today, most shipped merchandise is usually prelabeled by the manufacturer and assigned Universal Product Codes (UPCs) to make identification relatively easy. Unfortunately, despite our best intentions data "shipments" don't bear such intrinsic, uniform, and consistent labeling—mainly since our data warehouses aren't exactly brimming with meaningful and consistently assigned data element names. Is it any wonder that imposing rigorous data management principles is a lot like fighting Conan the Barbarian with only harsh words as weapons? The fact is, everyone from junior programmers to senior analysts has jumped into the data-naming act. Critics of the resulting pluralistic paradigms suggest that a need for alternatives exists, but although they talk a good game, few will step forward to cast the first stone. And although CASE tools can help, they're useful only if suitable guidelines are also deployed.

Clearly, if our primary objective is accurate classification and consistent naming of data elements, it's imperative that we properly

identify the "goods," which, in our case, is data. As usual, getting to this objective can be half the fun, but because the first step is often big, this column will suggest a method for cutting it down to size. First, we'll select a data-naming structure and assign the role of domain classification to part of it. Then we'll affirm the importance of rule-based domain analysis, observe its correlation to scalar measurement, and uniformly assign a class word to the data name based on that scalar/domain correlation.

Most data processing practitioners agree that consistently named data elements can simplify data design and management. One evolving standard is the Information Resource Dictionary System (IRDS) data element-naming paradigm, which incorporates a three-part name consisting of one class word, *n* generic modifier(s), and one prime word. The order of the name components can be rearranged. Figure 1 shows a product distributor's possible data element names for markups.

As structures go, the IRDS paradigm is sound and workable.

However, while data element names may be structured uniformly, their component words are usually assigned subjectively (sometimes capriciously) based on personal perceptions and predilections, rather than done objectively through informed analysis. The IRDS paradigm represents good form, but it still needs substance.

Recall that a domain is the set of allowable (or valid) values for a particular data element; that is, every data element must have a domain defined over it, but a particular domain value doesn't have to appear in the database to be valid. Two tables can be joined if each contains a column that is defined over the same (atomic) domain or a subset thereof. However, as in our example, different domains usually embody different properties, relationships, business rules, and so forth. Therefore, domain symbols may intersect even though their underlying values aren't defined over the same real domain, and no subset relationship exists.

The phrase "the data is the business and the business is the data" may suggest the preeminence of domains: At a minimum, comprehensive guidelines for logical data representation—often as complex as the business itself—should embody some semblance of domain sensitivity (or semantics), consistency, clarity, and concision. Adopting the aforementioned generic modifier and prime word as a combined "domain word" can be useful in representing both enterprise-related entities and properties

Class Word + Generic Modifier + Prime Word			Domain
PCT	NY_RETAIL	MKUP	(5, 10, 15, 20, 25%)
PCT	LA_WHLISL	MKUP	(10, 20, 30, 40%)

FIGURE 1. Data element names for markups.


```

graph TD
    Root[ ] --> Qualifying[The element describes a  
QUALIFYING PROPERTY  
or attribute which can  
be measured using a(n)...]
    Root --> Quantifying[The element describes a  
QUANTIFYING PROPERTY  
or attribute which can  
be measured using a(n)...]
    Qualifying --> Nominal[Nominal Scale]
    Qualifying --> Ordinal[Ordinal Scale]
    Qualifying --> Rank[Rank Scale]
    Qualifying --> Interval[Interval Scale]
    Quantifying --> Absolute[Absolute Scale  
(Count of like objects)]
    Quantifying --> Ratio[Ratio Scale]
    Quantifying --> Conditional[Conditional  
The element describes the  
record itself]
  
```

The diagram is a flowchart on a light green background. At the top, a horizontal line with a central vertical line descends into two arrows pointing to two boxes. The left box is titled 'The element describes a QUALIFYING PROPERTY or attribute which can be measured using a(n)...'. From the bottom of this box, four arrows point to four boxes: 'Nominal Scale', 'Ordinal Scale', 'Rank Scale', and 'Interval Scale'. The right box is titled 'The element describes a QUANTIFYING PROPERTY or attribute which can be measured using a(n)...'. From the bottom of this box, three arrows point to three boxes: 'Absolute Scale (Count of like objects)', 'Ratio Scale', and 'Conditional The element describes the record itself'. Below these boxes, there are two rows of text. The first row contains: 'NME ADR ID NUM', 'CDE', 'CAT CLS SEQ', 'TIM DTE TMS', 'AMT SUM QTY TOT', 'PCT AVG', and 'IND'. The second row contains: 'TXT TYP FLG', followed by empty space, followed by 'DUR'.

NME ADR ID NUM TXT TYP FLG	CDE	CAT CLS SEQ	TIM DTE TMS	AMT SUM QTY TOT DUR	PCT AVG	IND
-------------------------------------	-----	----------------	----------------	---------------------------	---------	-----

through which values meaningful to the enterprise are communicated. How the enterprise chooses to communicate these values through the domain word is discretionary. Nevertheless, because data elements are domain-based (read domain-constrained), they must adhere to rules of consistency and integrity.

It's probably not a coincidence that identifying business rules to achieve domain discipline closely resembles the popular definition of measurement: A measurement is a rules-based assignment of symbols to the perceived degrees (or "extensions") of a property. In both instances, empirical properties are learned, interpreted, and formulated into consistent and meaningful rules. More specifically, domain analysis is a process that includes learning the business rules that govern the relationships between the domain's values. In short, domain analysis entails learning the domain's properties. Metrologists have named the mandatory structural similarity between a property and its underlying symbolic representation

Once we agree upon and assign the prime word and generic modifiers to a data element name, the only task that remains is to select the proper class word. If a class word represents a domain classification, how do we classify domains? Despite domain analysis followed by accurate "domain word" assignment, consistent domain

At a presentation given by Judith Newton of the National Institute of Standards and Technology, I was pleasantly surprised to receive a chart entitled "Taxonomy for Formation of Data Entities." However, I noted, and was greatly saddened by, the absence of an actual framework for assigning class words. Furthermore, I was informed that the ANSI classing committee (X3L8) hadn't decided on a method for data classification.

If scales can measure real properties, why can't they measure the representation of some property or some attribute's assumable values—namely, its domain? We already know that when we examine a domain we also need to determine the relationships among values. All that remains is to choose the appropriate “level” of measurement (as scale types are sometimes called) and *voilà*—we have a systematic method for forging consistent and meaningful data element class words from the principles of scalar measurement. Domain classification notwithstanding, scalar mapping can also be a useful precursor to domain-support analysis, which consists of identifying valid comparison operators, doing integ-

Nominal	A list of names, numbers, IDs, text, or properties that are neither measurements nor orderable; “nominal” refers to “name.” Examples: marital status, social security numbers, ZIP codes, phone numbers, color.
Ordinal	Any convenient relative property measurement or comparison scheme; permits relative ordering of a property, but the ordering lacks a natural origin. Examples: street numbers, mineral hardness (1=lead . . . 5=tin . . . 10=steel).
Rank	Same as ordinal scales, but the properties must also have natural boundaries. Examples: military ranks, automobile sizes (subcompact, compact, mid-size, full-size).
Interval	Measurement relative to an arbitrary origin. Examples: calendar dates, military time, number line.
Absolute	Measurement of discrete variable; a count or enumeration of like units. Example: head count.
Ratio	Measurement that has an origin (usually zero), an ordering, and a set of operations; all measurements are expressed as a ratio of an arbitrary unit. Examples: length, weight, volume, mileage, duration.

MARCH 1992

rity checking, and so forth.

Naturally, the two assumptions here are that class word usage is indeed domain-based and that any domain can be mapped to a commonly accepted scale type. Furthermore, a nontrivial caveat exists regarding the capacity of a database to accommodate changes—especially business rule changes—with minimal structural disruptions. The database design process includes a thorough analysis of all domains, which are sometimes referred to as both the “fabric” of the database as well as the “glue” that binds its structure. Doing any part of this design in a vacuum weakens it and becomes a formula for database instability. In particular, if an unforeseen business rule change requires domain redefinition, a new scale may apply.

Figure 2 shows three primary data classifications that distinguish among qualitative, quantitative, and conditional data elements. Under each primary class are the applicable scale types and the valid class words that they can assume. See Figure 3 for a description of each

scale type. Note that other scales exist, such as probability, binary, and so forth, but many of them may be deemed as variations of these more primitive scales. The class word “IND” (conditional) is for a data element that describes the physical record in which it appears; that is, a record indicator. Typically, this kind of data element is used solely to control application processing rather than to describe a real property or attribute of an entity occurrence.

Correlating class words to scales should be as intuitive as possible. Class words for industry-specific data can also be used (for example, “WGT” [weight] is used in the shipping industry and “DUR” [duration] is used in broadcasting). When considering what the underlying values in the data element’s domain represent and how they’re used, it’s important not to confuse domain order with symbol orders. For example, the domain order of all integers between zero and nine inclusive is (0,1,2,3,4,5,6,7,8,9), whereas the symbol order (alphabetic) is (eight,

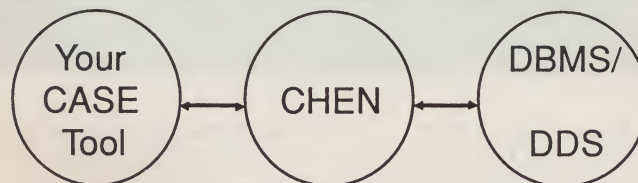
five, four, nine, one, seven, six, three, two, zero).

Keep in mind that measurement levels are cumulative with respect to the properties they can measure. That is, a higher measurement level such as an interval scale is more inclusive than a lower level such as ordinal. An interval scale can be adapted to measure an object or relationship exhibiting only ordinality, but an ordinal scale cannot be adapted to perform the same function as an interval scale.

Consistency is the basis of any useful taxonomy. Unfortunately, a useful *data* taxonomy is the exception rather than the rule. Nevertheless, it’s clear that a thorough data analysis (including domain) is a prerequisite for assigning data names consistently and objectively. As we’ve seen, applying key scalar measurements to domain analysis can provide a framework for systematically translating data element metadata into consistent and meaningful data names. ■■■

Robert Takoushian is the data administrator for a major communications firm.

Bridging Your CASE Tools to DBMS/DDS



You and your colleagues have been working very hard in the past few months to input the user requirements into your CASE tool only to find out that it does not talk to the file system, DBMS, or Data Dictionary Systems you already have. You may or may not have been involved in the original CASE tool selection, but it is too late to blame anybody now. Fortunately, CHEN has the solution!

Since CHEN Workbench can interface with more than 30 (Yes, 30!) commercial DBMSs and Data Dictionary Systems and quite a few CASE tools, it is very likely that CHEN already has it made for you. If not, we can develop it quickly to save your project from a disaster. So, if you are using IEW (or Excelerator,...) and want to convert the design data to Sybase (or Progress, Informix, DEC CDD+, CA IDD,...), give us a call. And you will be happy you did.

Also, if you want to convert from one file/DBMS to another or from one CASE tool to another, talk to us.

CASE-TO-CASE



DBMS-TO-DBMS



Chen & Associates
4884 Constitution Ave., Suite 1-E, Baton Rouge, LA 70808
(504) 928-5765 Fax (504) 928-9371
CIRCLE 13 ON READER SERVICE CARD

SQL Productivity Tools Every Step of the Way.

STEP 1

Design & Analysis

Deft

STEP 4

Operational Control

DBA Companion Environment
SA Companion

STEP 2

Application Development

SQL♦ Advantage Gateways
SQL♦ Debug 4GL Environments
SQR Family

STEP 3

Testing & Tuning

Performance Accelerator
SQL♦ Batch
TOP* Converter

SQL Productivity. Two words often promised but, until now, never delivered by a single vendor. That was before SQL Solutions stepped in with the SQL Productivity Environment, the first suite of multi-RDBMS SQL productivity tools for the entire SQL Application Lifecycle.

Armed with these tools, SQL Solutions consultants have designed, developed and implemented hundreds of applications for every major RDBMS, including SYBASE™, ORACLE®, Rdb™, INFORMIX™, INGRES™, and DB2™.

SQL Productivity Environment features:

- **Deft**, the premier RDBMS CASE Solution for Macintosh, VMS and Unix.
- **SQL♦ Advantage**, a programmer productivity environment for SQL.
- **SQL♦ Debug**, an interactive, source-level debugger for SQL.
- **4GL Environments**, high-performance 4GL application development tools.
- **SQR Family**, a family of 4GL report writing tools.
- **Gateways**, gateways services opening up RDBMS worlds (SYBASE, ORACLE, Rdb, INFORMIX, INGRES, RMS).

- **TOP* Converter**, a tool that automatically converts SQL*Forms 2.3 to 3.0 triggers.
- **Performance Accelerator**, an optimizer for ORACLE SQL*Forms.
- **SQL♦ Batch**, a tool adding multi-tasking to RDBMSs under VAX VMS.
- **DBA Companion Environment**, operational control tools for DBAs.
- **SA Companion**, tools for System Administrators.

SQL Productivity Environment is the most comprehensive suite of tools in the industry today. And backed by our SQL Systems Integration services, we're with you every step of the way. Call us. Today.



SQL Solutions, Inc.

The SQL Systems Integrators

CIRCLE 14 ON READER SERVICE CARD

8 New England Executive Park, Burlington, MA 01803,
617-270-4150 or 1-800-933-0044

SYBASE and Deft are trademarks of Sybase Corporation, ORACLE and SQL*FORMS are registered trademarks of Oracle Corporation, Rdb, RMS and VMS are trademarks of Digital Equipment Corporation, INFORMIX is a registered trademark of Informix Software, Inc., INGRES is a registered trademark of Ask Computer Systems, Inc., DB2 is a registered trademark of IBM Corporation, Macintosh is a registered trademark of Apple Computer, Inc., TOP*Converter is a trademark of Comtecno B.V.

CELKO ON SQL

Cursors versus queries, and COBOL versus SQL

"Even if you invent a story, you stick by it. We're journalists here!"

—Editor at Weekly World News to reporter Mark Kramer

"Even if you screw up a database, you stick by it. We're programmers here!"

—Joe Celko

A FRIEND OF MINE AT Nissan Motor Acceptance Corp. (NMAC) recently asked me to help with a database design problem. NMAC's loan applications are run against one or more credit bureaus—which credit bureau it uses depends on the ZIP code of the dealership or applicant. Obviously, it's better for people in New York to use a New York credit bureau rather than a California bureau. Each credit bureau has a two-letter abbreviation code and its field is a concatenated string of these two-letter codes, up to five codes in length.

NMAC's proposed method for handling this problem was to write a procedure in a host language that would accept the string as a parameter, parse it, and do all the credit checking work via leased telephone lines. The query for this method is simple:

```
SELECT *  
FROM CreditBureaus  
WHERE (zipcode = :ThisZip);
```

The question is, is this table properly normalized? The immediate response is no, because the string hides a one-dimensional array of two-letter codes. Arrays violate the flat file condition for first-normal form. Instead, the schema

BY JOE CELKO

Queries and COBOL

should have a table with the columns ZIP code and credit bureau code.

Upon further investigation, I found that the table of ZIP codes and credit bureau codes is keyed on the ZIP code, so it always produces the same vector for each ZIP code. Each credit bureau's information appears once in its ZIP code.

Encoding schemes are a mania of mine, as loyal readers will remember from an earlier column ("Make or Break Your System," DBA Shoptalk, March 1989). The credit bureau uses a concatenation scheme. However, two kinds of concatenation encoding schemes exist: ordered and unordered. In an ordered scheme, "AB" isn't the same as "BA," but they are the same in an unordered scheme.

The next question is, what does this order mean, if anything? I discovered that credit bureaus are ordered in the string from least expensive to most expensive to use. This order minimizes credit report costs. Therefore, if you normalized the data, you would need a table with the columns ZIP code, credit bureau code, and report cost. The query that will give you the least expensive bureau for a given ZIP code is easy. Just remember to re-

strict information to the target ZIP code as follows:

```
SELECT bureau#, cost  
FROM CreditBureaus  
WHERE (zipcode = :ThisZip)  
AND cost = (SELECT MIN(cost)  
FROM CreditBureaus  
WHERE (zipcode = :ThisZip));
```

However, this ideal situation isn't always the remedy. If you can't find information on an applicant at the first bureau, you'll then have to try the rest of the credit bureaus in order of increasing cost. Finding the n th cheapest bureau is a little harder, however. Assume that no two credit bureaus in the same ZIP code charge the same rates. If you ask for the $(n+1)$ bureau in an area with only n bureaus, then you'll expect to get back an empty result table. The host language placing the credit inquiry can use this empty table as a flag to stop.

You can also perform this operation with a cursor ordered by ascending cost. Use the cursor to load a five-element array and loop through the elements. However, keep in mind that cursors are frowned upon by pure SQL fanatics because they use a host language to perform.

THE BEST PURE SQL solution I've found is a self-join with a GROUP BY clause. The idea is to take each cost within a given ZIP code and build a group of costs that are less than or equal to it. The cheapest cost is in a group with one row, the second cheapest is in a group of two rows, and so forth. The n th cost will be the largest value in its group, so you can get a single-row

answer by pulling out the maximum cost for each group. The query can be changed by using a new value for *n* in the HAVING clause and a new ZIP code in the WHERE clause. The query follows:

```
SELECT A.bureau#, MAX(cost)
FROM CreditBureaus A, CreditBureaus B
WHERE (A.cost >= B.cost)
      AND (A.zipcode = B.zipcode)
      AND (A.zipcode = :ThisZip)
GROUP BY A.bureau#
HAVING COUNT(*) = (:n);
```

Notice that this query is exact: If two credit bureaus in the same ZIP code charge the same price, the results are empty. Using the BETWEEN predicate or modifying the HAVING clause to use "<=" can pull out any contiguous sublist.

Now, I should be honest: The cursor is probably faster and easier to use than my snappy query. SQL programmers are like a programming language (APL) programmers; they can write anything in one statement! And, just like APL, when they're finished, other programmers can't read it. The real problem with tricky queries isn't that they're hard to read later; rather, they can take days to run because they often have hidden Cartesian products and self-joins, operations on nonindexed columns, and such other CPU cycle-eating monsters.

RED BRICK SYSTEMS of Los Gatos, California was founded by Ralph Kimball, who also founded Metaphor Computer Systems. He is both Red Brick's president and technical expert. The company's product is a software SQL query optimizer called Gold Mine, which promises at least a 10 to one improvement in query speed. Therefore, this product should be able to help me with my killer queries.

Three versions of Gold Mine are available now. The stand-alone version is a small but complete RDBMS system in itself. The attached version works in tandem with an existing RDBMS server, while the integrated version places Gold Mine technology inside an RDBMS server.

Gold Mine is aimed at data-

A single SQL statement can literally equal several hundred lines of COBOL

bases over 10 gigabytes, so don't expect to see it on your laptop. Because it tries to avoid building scratch files, the required storage space is based on the size of the answer, not the complexity of the query. As a rough estimate, index generation is 15 percent of the original flat file size.

Because you have to do a lot of work to get the information that Gold Mine needs from the database, the company recommends schema tuning for best results. The product's weakness is that all join paths must be known at schema creation. As a result, only equi-joins can be used on foreign keys. The product works best with a star join; that is, one central table with columns that act as foreign keys to other tables. In the real world, these restrictions aren't bad for performance improvement.

The other part of the product is Reduced Instruction SQL (RISQL), which is an extended version of SQL-89. Any standard, vanilla SQL query can run in RISQL. The extensions are for aggregate functions that depend on the ORDER BY clause to let you do traditional break-and-summarize reports.

Overall, I don't like RISQL extensions within the SELECT clause because they're dependent upon an ordered result table that violates the set orientation of SQL. Many of the extensions are very complex and hard to figure out. Some of them will also be available in SQL2, but with different syntax. Red Brick Systems should have looked ahead and checked the forthcoming standards.

I can write all of the product's new aggregate functions in standard SQL. Although writing the functions requires horrible, complex CREATE VIEW and SELECT statements that run for days, it's possible. Remember my earlier remarks in this column about SQL program-

mers who can write anything in one SELECT statement?

I would rather have a straight report writer that uses a standard SQL query to access data. This function will let you save the query while you change the formatting and summary functions for different reports. Therefore, I recommend that you look for the Gold Mine engine, but ignore the RISQL extensions.

AS MUCH AS I HATE to admit it, most of the commercial industry is still writing programs in COBOL. However, a single SQL statement can literally equal several hundred lines of COBOL. Furthermore, the single SQL statement can run faster and be easier to read than an entire COBOL program. For example, try to write a COBOL program to perform the following query:

```
SELECT dept#, MAX(salary), MIN(salary), AVG(salary),
FROM Employees
WHERE (sex = "F")
GROUP BY dept#
HAVING (COUNT(dept#)>1
AND AVG (salary)<(SELECT AVG(salary) FROM
Employees));
```

When you try to format a readable report with this query, you'll see some of the advantages of COBOL as a host language for SQL.

Micro Focus Inc. in Palo Alto, California provides a PC version of COBOL that works with XDB Systems Inc., which is based in Laurel, Maryland. XDB's claim to fame is that its product can look like ANSI or DB2 versions of SQL. A developer can write a program on the mainframe in IBM COBOL with DB2 and move it to Micro Focus COBOL and XDB on the PC and vice versa.

This dynamic duo is working quite well. In fact, I watched this combination produce faster turnaround and generally better code at NMAC. The PC environment is more developer-friendly than mainframes, and parallel efforts are always faster. George Schussel of Digital Consulting Inc. in Andover, Massachusetts reported that one of his clients was able to transport a 130,000-line COBOL program from

the mainframe to the PC by changing only 12 lines of code—major league downsizing! I've been told that the UNIX version of Micro Focus COBOL works as well and allows easy movement from DOS to UNIX, but I have no experience with this product.

ACUCOBOL, WHICH also has a good reputation for portability and downsizing, has taken an approach other than providing embedded syntax. It has a general file access mechanism in its COBOL that looks like standard COBOL I/O to the programmer but, underneath this level, it hooks into specialized interfaces for different file systems. Its new Acu4GL product is an extension of this idea. It provides a link to SQL databases while hiding the details from the COBOL programmer. So far, it has Oracle and Informix links, with others in the works. (As an aside, I hate the product's name because it sounds like a 4GL product, rather than a database access tool. Instead, maybe the company should try "AcuAccess," "AcuDBlink," or something similar.)

Compiler directives in Acu4GL's COBOL file-definition code map structured records and SQL flat files into each other and handle datatype conversions, such as the conversion of numeric fields to date columns and renaming fields. This function means that group names and REDEFINES aren't allowed in the mappings. Acu4GL's two goals are to have no data duplication and let programmers write simple, straightforward COBOL without having to know anything about SQL.

Micro Focus COBOL is a true compiler, generating executable modules, while Acucobol is interpreted. This function explains how the company is able to change the I/O at run time and claim faster compile time than Micro Focus. The trouble is, you pay for this faster compile time in execution speed and code size for every run after the first one.

Ignoring the COBOL compiler differences, I'm a little hesitant about Acucobol because its SQL interface doesn't adhere to the

ANSI-defined host language interfaces. This inconsistency could be a handicap for porting mainframe applications to the PC and vice versa. Another weak spot is the product's lack of an indicator variable similar to those in the ANSI-standard embeddings. An indicator variable is an integer value that returns a flag when errors or NULLs in its associated variable exist. Instead, Acu4GL's NULLs are mapped to zeros or blanks.

However, the product offers a fast way to downsize existing,

older COBOL applications to a PC with an SQL database. Once you have mapped the SQL into COBOL, all of your code will port without changes.

If any of you have worked with these products, please let me know what you think of them in care of *Database Programming & Design*. I'll offer a new book as a prize for the best case study. ■■■

Joe Celko is a database-design consultant in Los Angeles. He is currently working as a consultant for NetBase in Torrance, California.

CDB Software Announces the APPLICATION ENABLING SERIES for DB2



CDB/REXX

full-function interface between REXX and DB2; the simplicity and power of REXX with the superior data handling capabilities of DB2

CDB/EDIT

"Back-to-Basics" DB2 Table Editing; efficient and intuitive

CDB/EASYSQL

SQL composer for DB2 programmers; step-by-step statement building and help facilities for the DB2 novice, time savers for the experienced programmer



CDB Software, Inc.
6464 Savoy
Suite 120, Dept. D
Houston, TX 77036
713-780-2382
FAX 713-784-1842

AE I•O•U: Send or FAX for information on the new AE Series of DB2 Tools.

Name _____
Title _____
Company _____
Address _____
Telephone _____

And sometimes Why: Call 800-627-6561, our new 24 hour support line.

CIRCLE 15 ON READER SERVICE CARD

(800) 227-4679 Dwayne MS

RESEARCH
(UNIX) (DB2) (VAX) FINANCE
(AS/400)

SQL
SERVER!!

(SQL SERVER)
DALLAS OFFICE

... WINDOWS NETWORK ...

Stephanie (APP. DEV.)
Joel (EIS)
Juliana (ORDER ENTRY)

Finally, a server that has no problem relating to others.

New Microsoft® SQL Server 4.2 is open to any kind of data.

In virtually any format. From any source, anywhere in your enterprise.

That makes it the only database server that can integrate all your information systems. Which, of course, helps your users get more information. And your company get more competitive.

SQL Server reaches out and talks to your minis and mainframes. On platforms as diverse as UNIX®, MVS, and VMS®. With products as diverse as Oracle®, SYBASE®, DB2™, and Rdb.

It's equally compatible with network standards like LAN Manager, NetWare®, VINES®, and LAN Server.

It can even handle non-relational data sources like VSAM files, CICS, and satellite information feeds – an ability beyond the reach of other servers.

All of which has an important result. As you add client-server technology, you don't sacrifice your existing systems. You simply improve them.

Dramatically.

Even better, SQL Server 4.2 inte-

grates seamlessly with the Windows™ operating system. So data isn't just easier to get. It's easier to use.

Developers, for instance, can now choose from over 120 tools – and use scrollable cursors that make creating Windows applications easier than ever.

And SQL Server has new, integrated Windows administration. So you can manage multiple servers from a single desktop.

Controlling data is easier, too. Our program-
mable server architecture lets you enforce data integrity, application logic, and

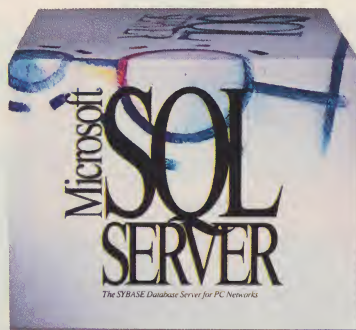
even business rules. Right at the server.

The net effect? You can finally integrate corporate data, manage it intelligently, and deliver it to your users.

Quickly, safely, and accurately.

So call (800) 922-3675, Dept. Y51, to order a customer solutions video, or to get a customized information pack on the new SQL Server 4.2.

We think you'll really relate to it.



The new Microsoft SQL Server 4.2 database server. It takes all kinds.

Microsoft®



*Client/server architecture lets MIS shops pursue the "best of breed":
Here's a guide to help you make sure the best fits with your breed*

A Client for Your Server

ONE OF THE MAJOR benefits of the client/server architecture is that it lets developers and users choose tools that best suit their needs. This "mix-and-match" approach is possible because application programs run on the "front end": powerful desktop computers independent of the "back-end" database server. Front-end development tools and applications communicate with back-end database servers through application programming interfaces (APIs), which are provided by DBMS vendors.

APIs are used by client applications to call library routines that transparently route SQL command requests from the front-end client application to the database server. The routines control all access to the database server and manage network access so that the applications developer need not be concerned with low-level network

protocols. Once the request is executed by the database server (via SQL commands such as SELECT, UPDATE, INSERT, and DELETE), the database server sends data records or messages back to the client.

This scheme makes application logic independent from server operating systems (such as OS/2, UNIX, VMS, MVS, and LAN Manager) and network operating systems and protocols (such as LAN Manager, Netware, TCP/IP, and Pathworks). Front-end vendors and developers can use this architecture to build tools that access a variety of database server platforms.

FORMS DEVELOPMENT TOOLS

Database server vendors usually offer their own set of forms development tools tailored for use with the vendor's DBMS. Examples of these would be Oracle SQL*Forms, Ingres 4GL and Windows/4GL, Sybase APT-Workbench, and In-

formix 4GL. Most of these development tools are designed to work solely with the vendor's DBMS. The exception is Ingres's 4GL and 4GL/Windows tools, which are capable of working with Tandem Computers' Non-Stop SQL and Digital Equipment Corp's Rdb/VMS as well as the Ingres RDBMS.

All of these tools have good functionality, but often users will have requirements that make third-party products a better choice. Some of these requirements may include:

- ☐ More functionality
- ☐ Better performance
- ☐ Support for a particular client platform (for example, Windows or OS/2 PM)
- ☐ Support for a greater variety of database servers
- ☐ Better portability among clients and platforms
- ☐ Better support for team development (for example, shared libraries)

□ Superior debugging aids.

Therefore, dozens of third-party forms tools are currently marketed for use with the major SQL database servers. The Microsoft/Sybase product, SQL Server, alone has over 100 front-end products from which to choose. Once you've decided to evaluate third-party tools, it's a good idea to set up some criteria for quickly eliminating potential products and developing a short list of contenders. Normally, the database server will be selected first, thereby eliminating front-ends that do not have links to the target server. The next step may be to decide on the client operating system platform.

CHARACTER-BASED VS. GUI

Client applications can either be character-based or graphical user interfaces (GUIs). DOS and UNIX workstations are popular character-based operating systems while Windows, OS/2 Presentation Manager (PM), Motif, or X-Windows represent possible GUI environments.

Character-based systems have been around for a long time and offer mature and proven capabilities. Examples of popular client/server development tools for UNIX and VAX database servers are Uniface Corp.'s Uniface, JYACC Inc.'s JAM, Unify Corp.'s Accell, and Cognos's Powerhouse. Borland International's dBASE IV and Paradox, DataEase International's DataEase, and Revelation Technologies' Revelation are examples of commonly used DOS front-end development tools.

Client/server computing is new, so look for products that have been used in product client/server environments. Just because a product works well in multiuser or PC network environment does not mean it will perform well in a client/server application. Many character-based systems were not designed for client/server and do not link well with the back-end server or take advantage of database server capabilities. The best way to judge a product's worthiness is to test it out and find references who have set up applications similar to yours.

GUI environments are new and do not have the user base or maturity of character-based products. Most of the recently released

Normally, the database server will be selected first

GUI front-end products have been designed specifically for client/server computing, making better use of desktop power through GUI interfaces, and also provide better support for database server advanced capabilities such as complex data types (for example, graphics, images, documents, and so on). Some of the choices for use under Microsoft's Windows 3.0 are PowerBuilder, ObjectView, Omnis 5, and SQLWindows. Application Manager and Object/1 are two products that work under IBM's OS/2 Presentation Manager. Ingres Windows/4GL is an example of a product that can run on a number of GUI platforms, including UNIX Motif (for UNIX workstations) and Windows 3.0.

Many people have had good experiences with GUI products, but most have found that the products entail a long learning curve. Programmers have to learn a new method of programming for a more complex environment. Programmers must consider many new elements in the GUI form design, such as window placement, interaction with GUI objects (radio buttons, push buttons, pop-up messages, and so forth), color schemes, and fonts. In a GUI environment, simple character-based processes, such as choosing the length of an input field, are time-consuming because proportional fonts make maximum field widths variable and difficult to calculate.

GUI environments (for example, Windows, PM, and Motif) themselves are new, which adds to the potential problems. Many users have experienced unpredictable program failures and networking problems when using Windows and other GUI environments. GUIs also require more power than character-based systems; therefore, they should be deployed on machines with fast processors (preferably a 386 or more) and a good amount of memory (4MB or more). Character-

based tools are more appropriate for smaller machines.

Making a choice between character-based and GUI-based tools is difficult. Character-based environments are proven, but they do not offer the long-term flexibility and power of GUIs. GUIs are newer, have less of a track record, and have long learning curves—but also have much greater strategic potential. In evaluating GUIs for your application, consider the following:

□ The nature of the application. Does it require a GUI?

□ Complexity. Is the application small enough to reduce the risks involved with using a new GUI environment?

□ Other types of applications with which it will interact. Is multitasking and data exchange necessary?

□ The long-term direction of the company with regard to GUIs.

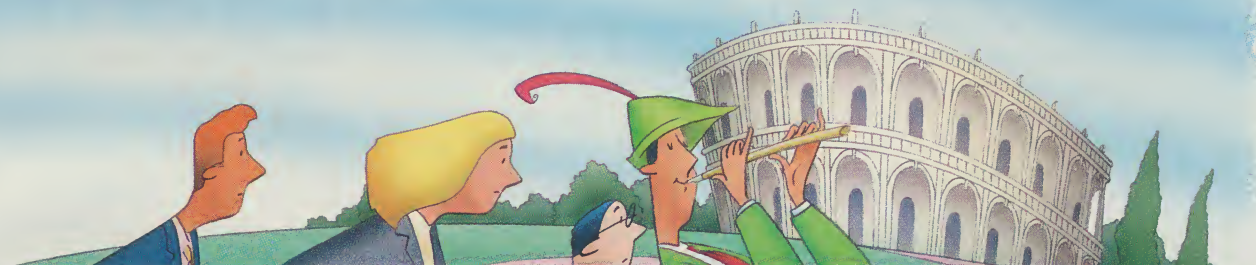
□ Existing hardware and network limitations.

If you need an application up and running quickly with the least amount of risk, you will probably lean toward a character-based forms tool. GUIs should be your choice if the application's requirements can be more easily satisfied in a GUI environment or if the long-term direction of your organization is moving toward GUI applications.

FORMS DEVELOPMENT

One of the major categories of front-end tools are those for forms application development. This software is used to generate the basic screens for maintaining a database. The rule of thumb is that if the DBMS vendor has a forms development tool, it will probably be the best fit for the back-end database server engine. For example, if the database server supports binary (or basic) large object (BLOB) fields—which can contain up to two gigabytes of data—the DBMS vendor's front-end tool will support access to this field. The DBMS vendor's front-end tool will probably support such specific database server features as backward and forward scrolling, special locking and concurrency controls, or stored procedures (partial transactions stored in the database server).

A third-party vendor may also support these back-end data-



YES. I want to know why UNIFACE is the only 4GL that does not ask me to pay the Open Systems Piper. Send me your free booklet "Guide to Building Future Proof 4GL Applications."

Name _____ Title _____

Company _____

Address _____ City _____

State _____ Zip _____ Phone _____

RDBMS/FMS _____ Operating System _____ Hardware _____

I would like to attend your free application development seminar.

City _____ Date _____

DP&D 3/1/92

S ARENA.

debugger, ability to
reporting tools and

openness cannot be added on or retrofitted, it must be there from the start.

We created UNIFACE 4GL to be open by design, from day one. Based on the ANSI ISO 3-schema architecture and industry standards, applications developed with UNIFACE are completely portable across DBMS/FMS, network, presentation interface, platform and CASE tool.

With UNIFACE 4GL, you can migrate applications from stand-alone to client-server, from character mode to OSF/Motif, Open Look, MS-Windows or OS/2 Presentation Manager, without rewriting a single line of code. You can also develop on SYBASE, ORACLE, INGRES, Informix, Rdb and a host of other DBMS/FMS across MS-DOS, OS/2, ULTRIX, Unix, VMS or VOS, and deploy on any or all of them without recoding.

you have a complete and open development environment.

We don't have to follow the "Open Systems" piper, we wrote the tune. Because we invented the "Open 4GL By Design" concept. More importantly, we delivered. And we have thousands of customers around the world who are glad we did.

Call for a copy of "Guide to Building Futureproof 4GL Applications", or sign up for a free application development seminar. Find out why UNIFACE is the only 4GL that is not asking you to pay the piper. **1-800-365-3608**

uniface

Advanced Software Technology

1420 Harbor Bay Parkway, Suite 140
Alameda, CA 94501-9781
415-748-6145 FAX 415-748-6150

OPEN 4GL BY DESIGN

UNIFACE is the registered trademark of Uniface International. The names of all other products mentioned are the registered trademarks of their respective companies.

CIRCLE 9 ON READER SERVICE CARD

Seminar Dates

March - May 1992

- ▼ March 10.....Research Triangle Park, NC
- ▼ March 25.....Parsippany, NJ
- ▼ March 26.....Boston, MA
- ▼ April 2.....Washington, DC
- ▼ April 14.....Toronto, Ontario
- ▼ May 14.....Chicago, IL



WE'RE NOT FOLLOWING ANYONE TO THE OPEN SYSTEMS ARENA.

A lot of vendors are now following the Pied Piper of "Open Systems", but will you be asked to pay the price? After all, openness cannot be added on or retrofitted, it must be there from the start.

We created UNIFACE 4GL to be open by design, from day one. Based on the ANSI ISO 3-schema architecture and industry standards, applications developed with UNIFACE are completely portable across DBMS/FMS, network, presentation interface, platform and CASE tool.

With UNIFACE 4GL, you can migrate applications from stand-alone to client-server, from character mode to OSF/Motif, Open Look, MS-Windows or OS/2 Presentation Manager, without rewriting a single line of code. You can also develop on SYBASE, ORACLE, INGRES, Informix, Rdb and a host of other DBMS/FMS across MS-DOS, OS/2, ULTRIX, Unix, VMS or VOS, and deploy on any or all of them without recoding.

Add to this UNIFACE's graphical 4GL debugger, ability to integrate text and data, friendly end user reporting tools and you have a complete and open development environment.

We don't have to follow the "Open Systems" piper, we wrote the tune. Because we invented the "Open 4GL By Design" concept. More importantly, we delivered. And we have thousands of customers around the world who are glad we did.

Call for a copy of "Guide to Building Futureproof 4GL Applications", or sign up for a free application development seminar. Find out why UNIFACE is the only 4GL that is not asking you to pay the piper. **1-800-365-3608**

uniface

Advanced Software Technology

1420 Harbor Bay Parkway, Suite 140
Alameda, CA 94501-9781
415-748-6145 FAX 415-748-6150

OPEN 4GL BY DESIGN

Seminar Dates

March - May 1992

- ▼ March 10.....Research Triangle Park, NC
- ▼ March 25.....Parsippany, NJ
- ▼ March 26.....Boston, MA
- ▼ April 2.....Washington, DC
- ▼ April 14.....Toronto, Ontario
- ▼ May 14.....Chicago, IL

UNIFACE is the registered trademark of Uniface International. The names of all other products mentioned are the registered trademarks of their respective companies.

CIRCLE 9 ON READER SERVICE CARD

base server features, but this support must satisfy three criteria:

- The special feature is accessible through the database server's API.

- The front-end tool is capable of using the feature (many tools developed before client/server cannot accommodate such features as BLOBs).

- The front-end vendor has made the resource and time effort to design the feature into the front-end product so it has practical usefulness; for example, the feature is fully supported, stable (does not crash), and performs well.

A few specific areas should be evaluated for efficiency and correctness of implementation:

- **Locking and Concurrency Controls.** Each database server has its own unique brand of locking and concurrency control mechanisms. For example, the Microsoft/Sybase SQL Server supports repeatable read and program controlled optimistic locking; that is, the front-end tool must check whether other transactions have updated the same records. However, like DB2, it supports an update through a cursor. Oracle defaults to read-consistency mode, which does not lock and record, so the front-end tool must do its own checking before updating any records. Gupta Technologies' SQLBase supports locking similar to DB2's, as well as a special type of read-only locking, while Novell's Netware SQL supports only table-level locks when the transaction is under Netware's transaction management (that is, COMMIT/ROLLBACK logic is being used).

The front-end tool must adjust itself for all of these possibilities, depending upon which database server is being accessed. Some front-ends support only one locking mode, while other front-ends support all of those available for the back-end servers. Often, a front-end will work particularly well with one of the database servers (the one to which the front-end vendor is most committed), while it will have less than optimal support for other database servers.

Some front-ends make locking transparent to the programmer, and others require the programmer to control locking directly. Programmer-controlled locking can lead to

The front-end tool must adjust to all of these possibilities

inconsistency of application implementations and could possibly lead to corrupted data. User-controlled locking limits the product's usefulness for users who are not trained in managing database locking. The best front-end tools mask the concurrency control completely from users.

- **Messages and Codes.** Front-end tools may replace database server-specific messages and codes with their own messages and codes. This feature means that you can migrate applications from one database server to another more easily. It is also important to have access to the database server-specific codes for debugging purposes.

- **Handling SQL and Extensions.** Each front-end tool must make decisions on the type of SQL commands it will generate to access the database server. For example, in some cases it may be better, for performance reasons, to issue an SQL SELECT command with a join and, in other cases, it may be better to avoid join logic and issue separate SQL SELECT commands for each table being accessed. Join operations themselves sometimes have to be optimized by the front-end tool because the back-end server lacks optimization. Oracle and Netware SQL, for example, do not implement cost-based (statistical-based) optimization techniques. For these servers, the front-end must analyze the database structure and try to determine the correct join syntax (for example, order of table names, whether to use an ORDER BY, and so on) to improve performance. Often, this analysis is not possible because insufficient data is available, and performance will be very poor.

The forms tool must handle nonstandard SQL commands, such as outer joins and null values, and then make two decisions:

- Whether the back-end supports the command (such as an outer join). If not, then:

- Use special techniques to emulate the necessary function. The software must be able to retrieve results in a consistent manner no matter how the database vendor has implemented the operation (for example, outer joins, null value operations, and so on).

The forms tool should also be able to use the database server's extended SQL language that includes procedural logic if one is available (examples are SQL Server's Transact-SQL, Oracle's PL/SQL, or Ingres/4GL). Similar support should be available for any extended built-in functions (for example, math, statistical, data/time, and decode functions).

Another area to look at is datatype support. Products such as the Microsoft/Sybase SQL Server and Interbase have BLOB fields that can contain up to two gigabytes of data. Every database server has its own set of datatypes, which may differ from the ANSI SQL datatypes. Each database server has its own implementation of time and date data types, long integers, floating point numbers, and so forth. It is important to investigate how well the front-end tool maps its own internal datatype support to the back-end server's.

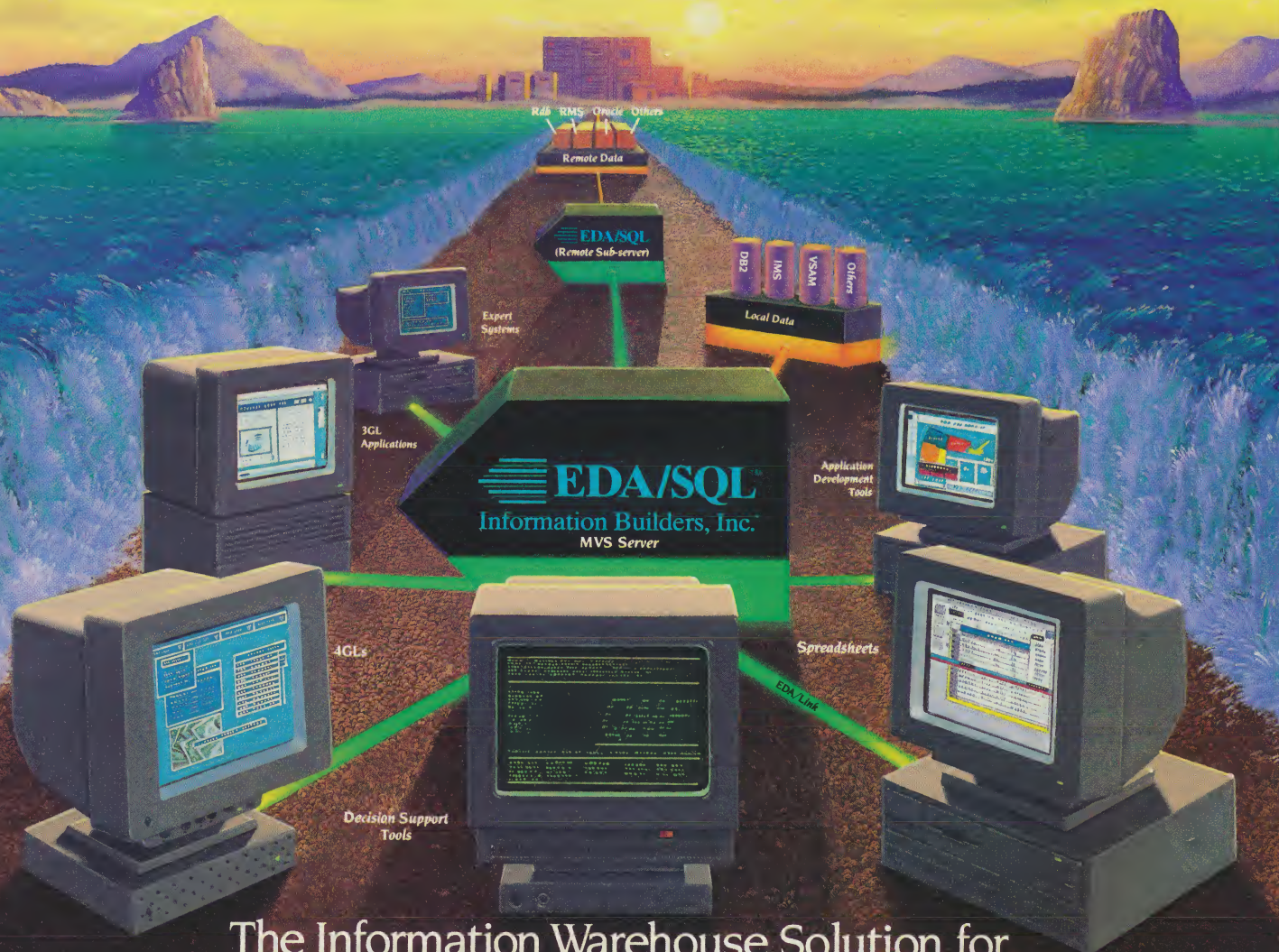
In some cases, the front-end software will allow access to extensions only if the programmer drops into a third-generation language (C, COBOL, and so on) procedure. This restriction affects productivity and the developer's ability to make maximum use of the front-end tool and the database server's extended features. If you purchase a database server for specific, nonstandard features, make sure all of those features are accessible from the front-end tool's basic facilities.

- **Processing Result Sets.** Forms tools can retrieve either a single row or sets of rows from the database tables, depending upon the SQL command issued. The retrieved rows are called the result set. If the result set is large, the database server retrieves enough rows to fill the workstation buffers and waits for the front end to request more rows from the result set.

The amount of local buffering can sometimes be controlled

JUST WHEN YOU THOUGHT IT WOULD TAKE A MIRACLE
TO ACCESS ALL YOUR CORPORATE DATA...

INFORMATION BUILDERS INTRODUCES ENTERPRISE DATA ACCESS/SQL™



The Information Warehouse Solution for SQL based Access to Relational and Non-relational Data.

The exciting breakthrough is here... Enterprise Data Access/SQL (EDA/SQL), the key component of IBM's Information Warehouse framework. EDA/SQL is a family of client/server products that works together to extend the reach of SQL based tools and programs for accessing both new data and legacy data stored in IMS, VSAM, and other non-relational DBMSs and files. EDA/SQL is the only product that gives you a uniform, relational view of data, regardless of how and where its stored.

From spreadsheet applications to complex business systems, EDA/SQL provides the power and flexibility to support

virtually any SQL based application. Popular tools like Lotus 1-2-3 and QMF - even 3GL applications - can use native commands and syntax to access over 45 local or remote proprietary databases and files.

Now, you can have a true open network architecture. Integrate new and existing hardware and network configurations with EDA/SQL's interlocking communications components that support most major network architectures. You can design the most cost-effective environment of interconnecting PCs, workstations, midrange and mainframe platforms. And have complete control over the data access environment

See us at DB/Expo '92, Booth #818

with centralized security management, on local and remote servers. You don't have to wait for a miracle. Now your data access problems can be solved. Call Information Builders today.

 **EDA/SQL**
Information Builders, Inc.

1250 Broadway, New York, NY 10001

800-969-INFO

212-736-4433; In Canada 416-364-2760
This advertisement refers to numerous products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies.

CIRCLE 22 ON READER SERVICE CARD

by front-end tool parameters. The parameters let the developer optimize local buffering for the kind of processing demanded by the application. Some front-end tools require that the full result set be retrieved before processing can continue. This requirement is a problem for transactions that process large result sets, because users must wait for all rows to be retrieved, and result set data must be stored in temporary local files, impacting performance and user response time.

It is important for applications to use any available backward- and forward-scrolling features efficiently. Backward and forward scrolling may be offered within locally buffered rows or for the entire result set. If a database server does not offer these features or if the forms tool supports these mechanisms, the application must retrieve the full result set and reposition itself within the result set whenever it scrolls backward. Reexecuting an SQL command and rereading a result set is very time-consuming and resource-intensive, so it should be avoided whenever possible. A good forms tool will have a customized interface, which will use a server's scrolling capabilities whenever possible.

□ Using Stored Procedures. Stored procedures are a method of storing complete or partial transactions in the server. Stored procedures reduce network communications overhead by minimizing client-to-server interactions. Triggers are specialized stored procedures that execute whenever a database table is updated or an event occurs in the database. The forms tool should also be able to access messages and return trigger codes whenever they are activated by the database server.

Forms tools should be able to use stored procedures from the application whenever they are available. Some forms tools, such as Uniface, not only let users access stored procedures but will automatically generate stored procedures and triggers to increase performance and enforce referential integrity rules. This task is transparent to the applications developer.

The method for executing and interacting with stored proce-



dures differs from server to server. With some servers, stored procedures are executed with an SQL command. Other servers use a 3GL call-level interface. Depending upon the form tool's capabilities, it may be able to interface to one or both of these types of stored procedures.

QUERY TOOLS

Query and decision-support tools are primarily concerned with retrieving and formatting data (as opposed to updating data). They must have well-designed interfaces, good functionality, and good query optimization. As with forms tools, it is important to decide whether the application will be in a GUI- or character-based environment.

At this time, only a limited number of client-based query and report writers use Windows or other GUIs. One of the more popular products is Quest from Gupta Technologies. Quest is an integrated query/reporting tool that easily integrates with Gupta's SQLWindows forms development tool. Quest can access Gupta's SQLBase database server directly through SQLBase's API, or it can access other database servers (for example, SQL Server and DB2) using Gupta's SQLNetwork routers. SQLNetwork routers are an additional per-workstation cost, so if the application will be accessing a database server that needs routers, this additional cost should be factored into the product evaluation.

Another popular GUI product is Channel Computing's Forest & Trees, which is also available in a character-based version. Forest & Trees accesses both SQL databases and other types of file formats, such as .DBF (dBASE) and ASCII files. Forest & Trees is primarily designed to monitor events in a database and signal the user that

the event has occurred; however, it does have a report writer that can be used for simple reports.

Quest's query and report writers are designed for users and developers and are capable of handling simple and complex problems. Forest & Trees is designed more for end-user monitoring and reporting and can access a variety of databases and file formats. However, it does not have Quest's breadth of reporting capabilities. In many cases, it is appropriate to use these tools side-by-side in a Windows environment.

Most Windows forms development tools do not have companion report writers. If you choose one of these tools, it may be possible to use products like Quest or Forest & Trees to complete the application, or it may be necessary to use a character-based query/report writer running in a DOS window. This arrangement may be awkward for users and the application will lack a seamless GUI feel, but it may be the only alternative available. If a character-based product is being evaluated, make sure it is thoroughly tested under Windows to ensure that functionality and stability are not lost.

A good source of character-based report writers are DOS DBMS products with SQL database server links such as Paradox, DataEase, dBASE IV, and Revelation. The query and report writing tools that come in these PC DBMSs are normally used in conjunction with companion forms tools that are packaged into the product. In comparing these tools, some areas you may want to compare are:

□ Ease of Connection. Connection and set-up procedures should be simple and as transparent as possible to the user. User interaction with database servers should be the same as working with native file formats.

□ SQL Support. Most of these products support a direct SQL pass-through mechanism. This device lets users enter an SQL command and send it directly to the database server. Developers need an SQL pass-through to test out SQL commands that will be included in an application. Users may need this facility to enter complex queries that can only be expressed in the

SQL command language.

□ **Query Interface.** Users generally prefer to avoid the SQL command language, if for no other reason than it is cumbersome to write out complete commands for every query. Most products offer query tools that shield users from SQL for simple and modestly complex queries (certain types of queries may be too complex to generate from the query interface). Query tools can be classified in three groups: query-by-forms, query-by-example, and prompted query.

Query-by-forms displays a form to the user that contains all (or some) of the fields contained in a table. The user enters values in the forms fields that act as the query criteria (that is, the AND conditions in the SQL WHERE clause). This approach is easy to use and works well if the user wants to see one record at a time. It is limited to queries that can be expressed with AND conditions (each field condition represents an AND condition in the generated SQL WHERE clause). It is difficult, if not impossible, to express OR or "nested" conditions (conditions that require parentheses) using this approach.

Query-by-example (QBE) type interfaces display field names across the top of the screen in a "table" format. The user enters conditions (examples) under each column name. Conditions can be simple or complex. Multiple tables are joined together by specifying "links" between example tables. QBE is used on mainframes and popular PC products such as Paradox. It can handle complicated queries (though there are limitations such as "nested" queries) and is relatively easy to learn. However, it is cumbersome to handle long records using this approach as opposed to the query-by-forms approach. Users must use left-right scrolling to enter criteria and view fields, while query-by-forms screens can display most or all fields on a single screen. The QBE interface is more difficult to learn than the query-by-forms approach, but this fact is counter-balanced by the increased functionality.

Prompted query interfaces generate SQL commands by asking users to choose tables, fields, join conditions, and other selec-



tion criteria from pop-up menus. Using this point-and-click approach, it is possible to create simple and complex queries—even nested SQL queries. It is relatively simple to learn, and users have an opportunity to view the SQL command as it is being developed. It takes more time to create queries using prompted query because of the number of menus that must be accessed to create one query.

Report writers also differ in capabilities and ease of use. Some report writers use a WYSIWYG interface to paint the report format, column headings, and field positions. Others use a command format, which directs the report writer with a series of format and input and output commands. The WYSIWYG approach is generally easier to use, while the command-driven approach offers more functionality. Quest is good example of a product that has excellent functionality and uses a WYSIWYG approach. SQR from SQL Solutions is a very popular report writer that uses a command approach and can access a variety of SQL database servers.

DECISION-SUPPORT TOOLS

More and more decision-support tools are available for database servers. Spreadsheets are commonly used as front-ends for SQL database servers. Microsoft's Excel uses a database connection product called Q+E to communicate with the servers. Quattro uses Borland's SQLink product. Lotus has two products available for linking 1-2-3 to database servers. Lotus Add-In is Lotus's original connectivity product. Add-In has been supplanted by DataLens, which is a more generalized and strategic product. Depending upon the database server, you may have a choice between using DataLens or Add-In, or you

may have to use Add-In only. All of these products work well, and the choice will depend upon which spreadsheet your organization decides to standardize.

Quattro has some advantages in that it uses SQLink (the same link that Paradox uses to access database servers) and can exchange information with Paradox. In this respect, Quattro offers a nice, seamless, workplace environment. All spreadsheet products should be evaluated for their ease-of-setup and how well they can optimize SQL commands for the target database. If users will be updating the database from the spreadsheet, the spreadsheet software link should be evaluated for how well it uses the database server's transaction management and integrity features.

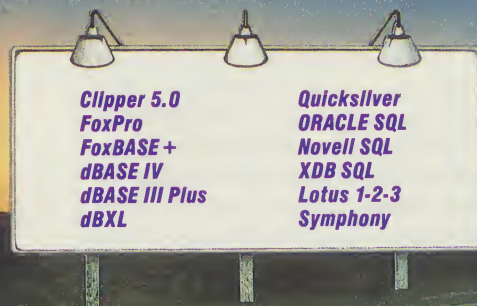
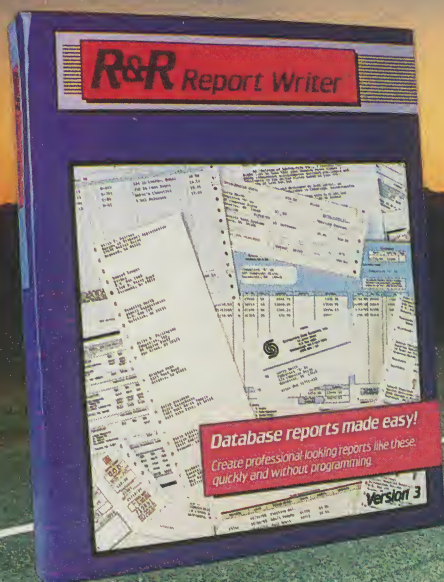
GETTING STARTED

Database server vendors usually keep a list of front-end products that support their servers. This list is a good place to start when looking for front-end tools, especially if you are not sure what is available in each category.

Front-end tools for database servers are maturing rapidly. Vendors have learned how to better optimize their products for various database servers. It is important to test out all products thoroughly under conditions that approximate production loads. Client/server products tend to have more problems under heavy loads, and communications links have a tendency toward breakdowns under stressed situations. Behavior varies from one front-end product to another and is very much dependent upon the database server and network facilities being used.

If you cannot do your own stress testing, the next best thing you can do is find references who are using the tools in an environment similar to yours in hardware, software, and network. This input will be invaluable for deciding how many users can be supported concurrently and what types of applications can be developed. ■■■

Richard Finkelstein is president of Chicago-based Performance Computing Inc., a consulting firm specializing in client/server product selection and applications deployment.



Drive down the road to great-looking reports with R&R.

Having a fast and efficient client-server engine will only get you so far. You need the right front-end to get at your data as well. R&R lets you create the reports you need in a fraction of the time it takes now, and with a fraction of the effort.

For people who program and those who don't.

From simple columnar listings to mailing labels to form letters to complex multi-page invoices, R&R helps you create the professional-looking reports you want. Without writing a single line of code.

Developers can use R&R's power and flexibility to avoid report programming and quickly reduce report backlogs. And R&R's royalty-free runtime allows reports to be built into applications and distributed to other users.

End-users and others who don't program can be immediately productive creating their own ad hoc and custom reports with R&R's simple, menu-driven interface, without having to know any SQL.

Designed for a variety of client-server systems.

R&R Report Writers for Oracle, NetWare SQL, and XDB are available now. Other versions are on the way. There are also R&R Report Writers for a wide variety of non-SQL databases.

This simplifies your choice, because when you learn one you know them all. And that makes development, training, and support a lot easier.

"The undisputed reporting champ...."

PC World, July 1990

"...a powerful and versatile report writer that is equally useful as either a programming or end-user tool."

DBMS Magazine, February 1991

"...a must see for any application developer or information service department."

InfoWorld, May 1991

"The first time I used it, R&R paid for itself 10 times over." "I don't know what I'd do without R&R." "Best product I have ever purchased in my life."

Unsolicited Comments of R&R Users

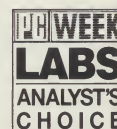
Try R&R. Join more than 100,000 others who already benefit from its time-saving features. Give your engine the front-end it needs. At only \$299, with a full 30-day money-back guarantee, you can't lose.

Call us NOW:

1-800-325-9035

Concentric Data Systems, Inc.

110 Turnpike Road, Westboro, MA 01581
508-366-1122, Fax 508-366-2954



R&R Report Writers are consistent award winners.

R&R Report Writer®

The Professional Strength Report Writer.

CIRCLE 23 ON READER SERVICE CARD

Responding to calls for greater user empowerment and the pressing need for better price/performance ratios has MIS professionals exclaiming, "Oh my!"

DOWNSIZING IS OFTEN portrayed as replacing mainframes with microcomputers, but a moment's thought quickly tells us that there's much more to downsizing than mere substitution. Picture a multimillion dollar IBM 3090 with 256 megabytes of memory, 200 gigabytes of disk storage, and 5,000 terminals connected to it. Someone comes in and disconnects the disk drives and terminals. A fleet of forklift trucks removes the mainframe cabinets (remembering to turn off the water) and wheels in a PC, to which the disks and terminals are then attached. It's a pretty ludicrous picture.

In fact, downsizing is only one result of the most important change ever in the way computers are used: the migration of applications from centralized computers to networks of distributed computers. This move is a consequence of a new way organizations use data, coupled with a change in computing technology. Each change reinforces the other.

THE PC FACTOR

Mainframe data was originally used for the company's operational systems: accounting, payroll, inventory, or personnel. Even after data entry shifted to an online environment, most of this data was still used in a batch mode for tasks such as reports or check writing. Although development of the mini-computer moved this business processing out to departments and branches, data processing, for the most part, has remained batch-oriented.

The advent of the PC dra-

Lions, Tigers, and Downsizing

matically increased the analytical uses of the computer, and significant amounts of data were moved to decentralized locations to facilitate this analysis. Unfortunately, the data quickly became outdated, and demands for access to the operational data could not easily be met without hurting the performance or risking the integrity of many of the central mainframe applications. Also, superior PC-based interfaces made users less tolerant of 3270 screens. Mainframe resources, however, were strained by attempts to make 3270 screens as interactive and user-friendly as those on PCs. This difficulty accelerated the shift to departmental computing, which increasingly relied on such computers as Digital Equipment Corp.'s VAX, IBM's AS/400, or networked PCs.

Another major motivation for decentralization was cost: Mini-computers and PCs were less expensive than mainframes. In part, the difference in purchase price was because the mainframe included a degree of service (such as on-site

systems engineers) that didn't come with the smaller computers. But an even larger factor was the degree of competition in the mini and PC arena that fueled rapid improvements in price/performance ratios, as well as the relatively easy availability of third-party peripherals. In fact, much of what is called downsizing is simply off-loading mainframe applications to distributed smaller mainframes and networks.

Up until the last few years, the quality of networks and interfaces limited the move to distributed, interactive systems. But recent improvements to products such as Novell's Netware 386 and Microsoft's LAN Manager, as well as the advent of distributed computing protocols, have corrected this problem. Add to this problem the increased presence of UNIX in the commercial world and more sophisticated graphical user interfaces (such as Windows 3.0 and Motif), and we have a critical mass necessary for greater migration.

Of course, obstacles still exist that make downsizing difficult. It

**PLATINUM
technology**



555 WatersEdge Drive
Lombard, IL 60148-9930
(708) 620-5000

Full
V2.3
Support

The Tools And Utilities For The DB2 Of Tomorrow. Tomorrow.

*available to thousands of locations across the U.S. †As of 12/91, support for DB2 V2.3 General Availability release of 10/24/91.
© 1992 PLATINUM technology, inc. All rights reserved. IBM and DB2 are registered trademarks and SystemView is a trademark
of International Business Machines Corporation. Federal Express and Priority Overnight are trademarks of Federal Express
Corporation and used with permission.

Start Exploiting All The New DB2 V2.3 Enhancements At 10:30* Tomorrow Morning.

Instead of delivering just a timely promise of compatibility with the new DB2® version, we're shipping full, enhanced support for DB2 V2.3† now.

To support the latest and greatest DB2 features, we've enhanced our entire suite of tools and utilities, making them up-to-the-minute and better than ever.

- ☒ Archive Log Command
- ☒ BINDAGENT Privilege
- ☒ New Catalog Tables
- ☒ Data Definition Control Facility
- ☒ Hybrid Joins
- ☒ Indexable LIKE :HOST
- ☒ Packages
- ☒ Package Conversion
- ☒ PLAN_TABLE Changes
- ☒ ROSHARE
- ☒ SQL - ALTER DATABASE
- ☒ SQL - CREATE TABLE ... OBID
- ☒ SQL - CURSOR WITH HOLD
- ☒ SQL - CONNECT
- ☒ SQL - DESCRIBE TABLE
- ☒ SQL - LIKE ESCAPE Parameter
- ☒ SQL - FOR xxxx DATA (SCBC/MIXED)
- ☒ SQL - OPTIMIZE FOR n-ROWS
- ☒ SQLSTATE
- ☒ SYSCTRL Privilege
- ☒ Threshold - Decimal 31 Digits
- ☒ Threshold - Index 64 Columns
- ☒ Threshold - Table 750 Columns

Our newest tool, PackageIt, will provide an automated package conversion facility and include a tutorial for all new V2.3 features.

When it comes to tools and utilities for DB2, we've never kept you waiting or wanting. So call us now at 1-800-442-6861 for V2.3 support and an airbill with your name on it.

It's the best way we know to make dramatic improvements in DB2 performance overnight.

**PLATINUM
technology**



PLATINUM technology, inc.

555 WatersEdge Drive, Lombard, IL 60148 (708) 620-5000

CIRCLE 24 ON READER SERVICE CARD



remains much easier to find mainframe COBOL programmers than good C programmers with client/server experience. The tools for managing networks are still in their early stages compared to the maturity of mainframe tools. And the difficulty of a mixed vendor environment with interfaces from PCs to local and wide area networks (LANs and WANs) to mainframes creates many opportunities for problems.

This article will examine the role of client/server computing in downsizing. We will consider what organizations can expect from a downsizing effort, and offer some suggestions on how to accomplish your goals.

MAINFRAME VS. PERSONAL

Let's consider once again the major differences between mainframe and personal computing. Mainframe computing is characterized by centralized, multiuser computing, in which the user interacts either through a batch job or a terminal. A systems administration staff's duties include software installation, data backup, and security. Personal computing, on the other hand, is characterized by decentralized, single-user computers, and users perform their own systems administration. Notice that these definitions are independent of the hardware platform. Thus, if you have a mainframe for your exclusive use, then that is personal computing. If your three-person workgroup is sharing an old IBM AT, then that is called mainframe computing.

Downsizing is based on a move to distributed systems in which client/server architectures are central. Client/server computing is a hybrid of mainframe and personal computing. The clients are single-user workstations administered for the most part by users, and are closest to the personal computing paradigm; whereas the servers are multiuser, centralized computers administered by a separate staff, and are closest to the mainframe paradigm.

Let's look more closely at the contrast between mainframe and client/server computing. In traditional host-based architectures, the database and applications run on a

Downsizing is based on a move to distributed systems

single host computer, frequently sharing the same memory. Mainframe computing is heavily oriented toward batch processing, using nonprogrammable workstations for inputting data and displaying the results.

The advantage of this approach is that it is easy to centralize administration and control the database. Furthermore, the technology for developing, deploying, and maintaining these applications is quite mature and stable. Applications run on inexpensive networks such as IBM's Systems Network Architecture, using 3270 data streams or asynchronous terminals. The amount of network traffic is relatively limited since only the actual data needed is shipped in either direction.

However, these host-based applications have a number of disadvantages. They require very powerful computers that support not only the DBMS but a multitude of applications and users. Each new application or user increases the demand for resources and quickly saturates even a large mainframe. Centralized control has also proved too unresponsive to the needs of users, particularly those who want to analyze the data or create ad-hoc reports. Furthermore, user interfaces of host-centered applications are often primitive compared to the best of the PC interfaces; and, because the interface is customized to a particular application, training costs are high.

The solution to these difficulties is based on distributing computing power and data closer to the ultimate users; in other words, using cooperative processing. In cooperative processing, portions of an application run on different computers, with data located at one or more sites. The simplicity of this definition hides two major issues: first, how the programmer logically divides the work; and second, finding a mechanism that

enables programs to work together efficiently.

CLIENT/SERVER

Client/server architectures, a subset of cooperative processing, address the first issue: the logical split of functionality. The user portion of an application runs on one or more client computers, and works cooperatively with the systems portion running on one or more server nodes. The user code running on the client generally handles functions including presenting and formatting data, data entry procedures, transaction logic, business rules and models, application flow control, reporting, and so forth.

Database servers are central to almost all downsized applications. The database functionality runs on a server, which provides the data dictionary (where database definitions reside), manages concurrency, handles journaling and recovery, regulates security, controls transactions, optimizes data access, and performs all the other database tasks. The data and DBMS may stay on the mainframe (which we would now call the server). But sometimes, because the application tasks have been off-loaded to the clients, a smaller computer can be substituted.

Some products that classify themselves as database servers are really providing cooperative processing—they let user application functions reside on the server. The mechanism most commonly used is called stored procedures. Stored procedures let the user write procedural logic containing SQL statements, name these procedures, compile them, and store them as part of the database itself. To perform this task, these products require extensions to standard SQL. SQL Server (from Sybase and Microsoft), Interbase (from Borland International), and Ingres (from ASK/Ingres) support stored procedures. The current releases from Oracle Corp. and IBM do not have this capability yet, but virtually all DBMS vendors intend to add it in the future.

One of the main benefits of client/server architecture is that it uses available network bandwidth effectively. In this arrangement, a high-level language (typically SQL)

is used to request the desired data. Rather than needing to ask for individual records, a single SQL statement can request data from many tables. The intelligence at the server can perform a significant amount of processing; therefore, rather than transferring entire files, only selected information is transferred. Local copies of data can be used for a variety of tasks, such as performing data entry validation (although this activity can raise some synchronization issues). Stored procedures reduce network traffic even more, since a complex procedure that already resides on the server is invoked by name instead of being sent to the server.

WHAT TO DOWNSIZE?

Although cost savings are the main goal of many organizations trying to downsize, they are difficult to obtain. We often read in the press of how an application on an older mainframe (commonly an IBM 4381) has been reimplemented in a networked PC environment at a significant savings of operating costs; however, these write-ups rarely give a complete accounting of the costs for the new hardware and software. Most cost justification analyses are good at showing where money will be saved from the old system, but are less accurate in estimating the costs of the new systems.

The real reason to downsize is to benefit from moving an old application that is clearly at the end of its life cycle to modern technology and a new design that reflects the organization's current needs. Cost savings are not so much the result of downsizing, but rather of taking an old application on older, more expensive hardware and reimplementing it on modern hardware. The cost of implementation will be lower than it used to be because of today's improved techniques, and the new hardware is far cheaper. Furthermore, the new implementation will be better at solving the current problems than the application it replaces.

Downsizing is frequently an incremental task, involving a specific application or group of applications that use a particular database. It is best to choose those applications for which the data

Client/server computing has its benefits as well as its risks

can be cleanly separated from the mainframe database; otherwise, you will have the problem of redundant data that must be kept synchronized. The easiest applications to downsize are those that can be localized so that they run on a single LAN of reasonable size, and the database can reside on a single local server.

Another important question to consider is how the data is used. Most likely, some applications will need access to remote data in addition to their own local data. Rather than create redundant data, it may be preferable to maintain a centralized DBMS that is accessed by these remote applications as needed. The choice between remote copies of data and centralized data depends largely on the frequency of updates, the number of updating sites, and the need to keep the copies of data synchronized. If the updates are frequent or from multiple sites, then centralization is better; if they are infrequent, then redundant data may be preferable. If the distributed applications can tolerate a period when they don't have the most current data, then the central site can function as a master database from which periodic snapshots are taken to update the subordinate databases on the LAN.

Given the present limitations of distributed database products, try to avoid those situations in which replicas of data must be kept in synchronization. Solutions to that problem are complex and involve a great deal of overhead.

Applications that can benefit from the documented productivity gains of a graphical user interface (GUI) are also good candidates for downsizing. Real evidence exists that GUIs can help save money. Their consistency makes applications easier to learn; GUIs also reduce training and support requirements. In addition, an improvement in quality comes from users being

able to create better-looking reports—frequently with graphics—due to the similarity between print and imaging models that facilitate WYSIWYG (what you see is what you get).

A constant concern for many mainframe users has been systems capacity, often necessitating a large, expensive step to the next size platform. In some cases, they even have to wait for the new, larger system to be designed and built. In a distributed environment, the incremental addition of users and applications puts less of a burden on the servers because much of the actual work is still being done on client computers. Therefore, applications that are running out of gas in a traditional mainframe computing environment are good candidates for downsizing. In fact, simply moving to a client/server architecture and retaining the mainframe as the server can extend the mainframe's usefulness.

Notice that these benefits do not result from moving to smaller computers, but from moving to distributed architectures in which the servers are sized for the application. While off-loading the client functions to a workstation reduces the size requirement for the server, the server must still be big enough to handle all the clients, the DBMS, and the portions of the application logic that run on the server.

DOWNSIZING CAUTIONS

While client/server computing has significant benefits, it is not without risks. Client/server computing is intrinsically more complex than centralized host computing. For example, if DB2 is the database server, a long chain goes from the client application to the server and back again. The query is typically formulated on the client workstation, travels across the LAN to a communications gateway, into a CICS transaction, then into DB2, and back again. Along the way, it may have to pass through mainframe security subsystems. The number of tools available for debugging problems in this chain is depressingly small, and the opportunity for finger pointing is very large.

In downsizing, it is impor-

tant to pick a suitable server. Some real questions exist about whether PC hardware is yet suitable for the multigigabyte applications common in the mainframe world. The scale of many such applications in terms of throughput, number of users, database size and complexity, transaction volume and complexity, and response time requirements is beyond the capacity of even the largest microcomputer servers. For such applications, the developer must seriously consider larger servers, such as a DEC VAX or an IBM 3090. On machines like these, memory can exceed 500MB, disk storage may be hundreds of gigabytes, and I/O bandwidths can be greater than 100MB per second. The downsizing difference is in the style of computing, not necessarily in the size of the server itself.

Another reason for looking at mainframes and minis as servers is that while a PC-based system may run out of steam as an application grows, more powerful servers are usually available in the lineup of mainframes. It's very clear that all the hardware vendors have recognized this fact. IBM and DEC, for example, are positioning their computers as servers in networked systems.

When choosing a server, the problem of the operating system also exists. MS-DOS is not adequate for most serious applications because of limitations in memory management and interprocess communications, which means choosing among OS/2, some version of UNIX, or a vendor's own operating system. Forthcoming operating systems such as Microsoft's Windows NT or Apple/IBM's "Pink" may also be suitable.

The future for PC DBMS applications, such as those built with Paradox, dBASE, and dBASE derivatives (such as FoxBase) should also be examined. As mainframe-type applications move to client/server architectures on PC networks using "industrial strength" DBMSs such as SQL Server, Ingres, Oracle, or InterBase, users of the PC DBMSs must consider whether to migrate applications to the database server. This consideration is particularly true for applications that are multiuser, have large data-

Gateways are an important part of the solution to the heterogeneity problem

bases, and use data from the server database. Vendors such as Borland International are developing interfaces to a variety of database servers. Therefore, it is sensible to consider migrating PC applications nearing the end of their life cycles to database servers, particularly if much of the front end can still be used with the server interface.

HETEROGENEITY WITH SQL

There is another more important reason why mainframes aren't dead: because data resides on them. Most essential corporate data resides in mainframe databases managed by DB2, IMS, CA-IDMS, VSAM, or any of a number of other products. The effort required to convert this data and the billions of dollars of applications that use it guarantees that data and applications will use this technology well into the next century.

Consequently, it is quite likely that a downsized enterprise will be moving to an increasingly heterogeneous environment, with multiple types of hardware, operating systems, networks, and DBMSs. In such a world, users and systems people quickly learn that there is far more to "open systems" than UNIX. Interoperability can be a very difficult problem to solve.

SQL is the Esperanto of the database world, but beware: Some vendors add SQL to a DBMS and pronounce it the solution to the interoperability problem. Unfortunately, complications occur. Not only are there differences among most SQL implementations, but other differences in the target DBMSs, such as the concurrency and locking models, may cause problems.

Gateways are an important part of the solution to the heterogeneity problem. A gateway takes the SQL for a particular DBMS, translates it to the appropriate language for the target DBMS, and

sends it to the remote computer, thus enabling client/server computing in a heterogeneous environment. The gateway must know the difference between the native DBMS and the new target. Some notable gateways include Micro Decisionware's Database Gateway, which lets programs using the SQL Server application programming interface access DB2 mainframe programs; Sybase's Open Server, which is used to build gateways not only to DBMSs but to remote applications; Information Builders Inc.'s EDA/SQL, which is the database access component of IBM's Information Warehouse; and the ASK/Ingres Gateways.

The implementors of distributed applications must also be conscious of the potential problems caused by security. While no one ever walked off with a mainframe disk farm, a PC-based server is distressingly portable. In addition, data is vulnerable to interception when transmitted. While physical protection of the data is a function of site security, the DBMS vendors have added significant security features, and some have conformed with U.S. Department of Defense requirements for secure databases.

Lastly, in a distributed architecture, data integrity must be cautiously guarded. As opposed to a centralized mainframe where the DBA group carefully ensures data integrity, a downsized environment has data residing in many locations, with updates coming from remote sites and from sources other than carefully constructed programs. In these cases, database triggers (procedures within the database automatically executed when an update occurs) are extremely important.

MANAGERIAL ISSUES

Testing and maintaining downsized applications may be much more difficult than with the mainframe versions that have been replaced. When a major new version of an application is brought up on a mainframe, a period of extensive testing is conducted before deployment. Also, new documentation must be written and users trained. Minor enhancements or fixes still require testing, but their documentation and training burden may be nonexistent.

A networked environment is not so straightforward. Suppose a mainframe application with 100 users has been off-loaded to four PC-based LANS. Now there are four servers and 100 client workstations. A new release must be tested on every server and client variation. To minimize this problem, it may be possible to ensure that the servers are all identical (although undocumented differences in so-called identical computers can in fact produce unforeseen problems). It is much more difficult to ensure that the clients are all uniform, including not only the hardware, but also the system software such as configuration and boot files. A number of organizations are using diskless workstations that boot from the server to minimize the likelihood of subtle differences among clients. Nevertheless, as time goes on, a small percentage of clients will probably require staff attention to update.

The physical burden of sending out updates can be quite large. Products such as IBM's Distribution Manager, Spectrum Concepts' Xcom/Software Distribution System, and Tangram's AM:PM are designed to simplify the release of new software across networks of clients and servers. Unfortunately, they require an MVS host from which the software is sent to networks. Still, a pressing need exists for improved tools to manage a downsized environment.

PROCEED WITH CAUTION

Beyond any doubt, the major trend in computing today is toward distributed systems of networked PC clients and PC servers, minicomputer servers, or mainframe servers. And because the benefits are quite real and significant, in the long run we will have better, more robust, and more usable systems. In the short term, however, we must proceed with reasonable caution and realistic expectations in order to realize the payoffs that are available. ■■■

Herbert A. Edelstein is a principal of Euclid Associates, a consulting firm specializing in database management and document image management. He is a consultant to computer vendors and users and is a frequent speaker and author on a variety of computer topics.

DATABASE

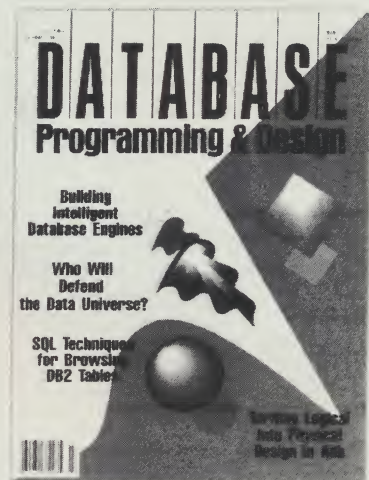
Programming & Design

***the first magazine to
focus on practical,
technical solutions
for DBMS specialists***

***—from the publishers of AI Expert,
Computer Language, UNIX Review,
and Embedded Systems Programming***

***Every monthly issue will
bring you practical, helpful
topics like these:***

• DB2 support tools • distributed query processing • location transparency • operating system independence • normalization • conceptual data modeling • structure and data integrity • security barriers • automated design tools • data migration • productivity tools • gateways • SQL dialects • real-time monitoring • data recovery • end-user interfaces



***No-risk charter offer for MIS
and Database Specialists***

GUARANTEE: after reviewing my first issue FREE, I may cancel if I do not wish to continue. If I do decide to continue, I may still cancel at any time and receive a full refund of my total subscription fee.

Send to:

DATABASE PROGRAMMING
& DESIGN, PO Box 53481
Boulder, CO 80322-3481

YES! Please put my name on your database as a subscriber to DATABASE PROGRAMMING & DESIGN, the magazine providing practical, technical solutions to DBMS specialists. I understand I will receive my first issue FREE and if I decide to continue, a full year's subscription for only \$37, a \$10 savings off the basic subscription price.

Note: Canadian delivery, US\$43. Other international, US\$52—please send payment with order in U.S. funds drawn on a U.S. bank.

Please check: ☐ Payment enclosed ☐ Send a bill ☐ Charge to:

☐ Visa ☐ MasterCard ☐ AmEx Card # _____

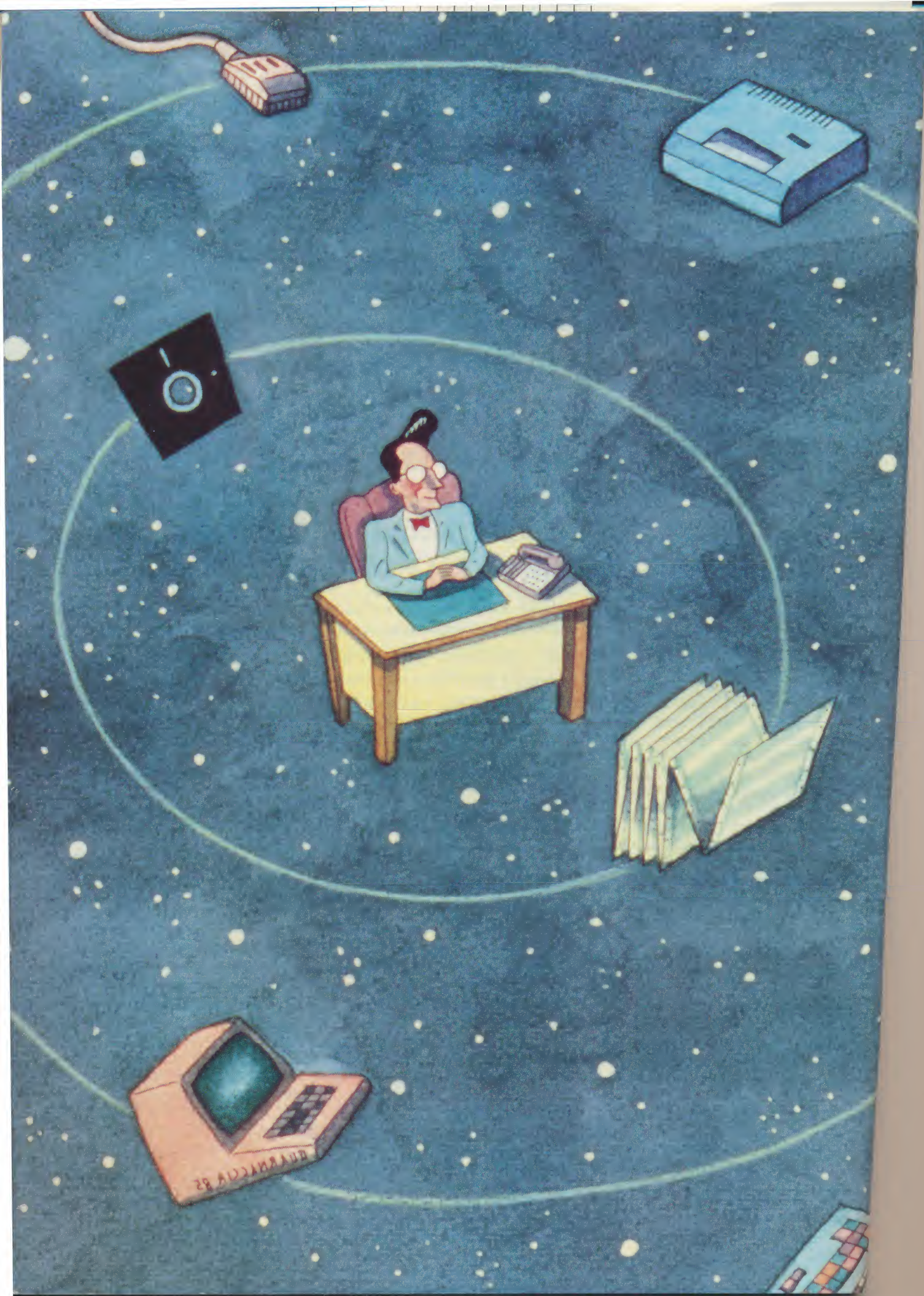
Exp. date _____ Signature _____

Name _____

Company _____

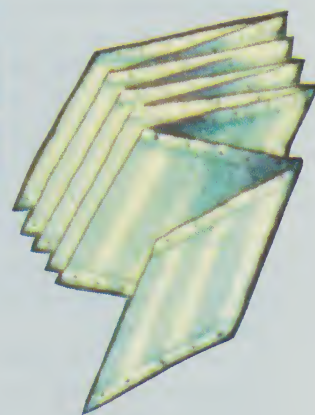
Address _____

City _____ State/Prov. _____ Postal Code _____



If they meet expectations, distributed DBMSs will replace centralized systems in many applications. But unsolved technical problems remain

Distributed Database Systems:



Where Are We Now?

DISTRIBUTED DATABASE technology is one of the most important computing developments of the past decade. During this period, distributed database research has been intense, culminating in the release of a number of first-generation commercial products. If it meets expectations, distributed database technology will impact data processing the same way centralized systems did a decade ago. Indeed, some observers claim that within the next 10 years most organizations will move toward distributed database managers, and centralized database managers will become an antique curiosity.¹

With the technology now at the critical stage of finding its way into commercial products, it is

important to seek answers to the following questions:

- ☐ What were the initial goals and promises of distributed database technology? How do current commercial products measure up to these promises? In retrospect, were these goals achievable?

- ☐ Have the important technical problems already been solved?

- ☐ What technological changes underlie distributed data managers and how will they impact the next generation of systems?

The last two questions hold particular importance for researchers because their answers determine the road map for research in coming years. Recent papers addressing these questions have emphasized scaling² and the introduction of heterogeneity and autonomy.³ These

problems are important, but many others remain unsolved. Even such much-studied topics as distributed query processing and transaction management involve research problems that have not yet been addressed adequately. Furthermore, new issues arise as the technology changes, application areas expand, and we gain experience with the application of distributed database technology.

WHAT IS A DISTRIBUTED DATABASE SYSTEM?

A distributed database is a collection of multiple, logically interrelated databases distributed over a computer network.⁴ A distributed database management system (distributed DBMS) is the software system that permits the management

ARTWORK: STEVEN GUARNACCIA

shareable. In fully autonomous systems, however, the individual components are stand-alone DBMSs, which do not know whether other DBMSs exist or how to communicate with them.

□ *Distribution* deals with data. We consider two cases: Either data is physically distributed over multiple sites that communicate with each other over some form of communications medium or it is stored at only one site.

□ *Heterogeneity* occurs in various forms in distributed systems, ranging from hardware heterogeneity and differences in networking protocols to variations in data managers. The important forms of heterogeneity from the perspective of database systems are differences in data models, query languages, interfaces, and transaction management protocols. The taxonomy classifies DBMSs as homogeneous or heterogeneous.

The alternative system architectures based on this taxonomy are shown in Figure 2. The arrows at the ends of the axes do not indicate an infinite number of choices but simply the dimensions of the taxonomy. This article deals mainly with tightly integrated, distributed, and homogeneous database systems.

CURRENT STATE OF TECHNOLOGY

Like any emerging technology, distributed database systems have their share of fulfilled and unfulfilled promises. In this section, we consider the commonly cited advantages of distributed DBMSs and how well current commercial products provide these advantages.

TRANSPARENT DATA MANAGEMENT

Centralized database systems have taken us from a data processing paradigm in which data definition and maintenance were embedded in each application to one in which these functions are abstracted from the applications and placed under the control of a server called the DBMS. This new orientation results in data independence—the immunity of application programs to changes in the logical or physical organization of the data and vice versa. Distributed database technology extends the concept of data

Most distributed DBMSs do not provide sufficient transparency

independence to environments in which data is distributed and replicated over a number of machines connected by a network.

Data independence is provided by several forms of transparency: network (and, therefore, distribution), replication, and fragmentation transparency. Transparent access to data separates a system's higher-level semantics from lower-level implementation issues. Thus, database users would see a logically integrated, single-image database, even though it was physically distributed, enabling them to access the distributed database as if it were a centralized one. In its ideal form, full transparency would imply a query language interface to a distributed DBMS no different from that to a centralized DBMS.

Most commercial distributed DBMSs do not provide a sufficient level of transparency. Part of the problem is a lack of support for replicated data management. Some systems do not permit data replication across multiple databases; systems

that do permit it require that the user be physically logged on to one database at a given time. Some distributed DBMSs attempt to establish their own transparent naming schemes, usually with unsatisfactory results, requiring the users either to specify the full path to data or to build aliases to avoid long path names. An important aspect of the problem is the lack of proper operating system support for transparency. Network transparency can easily be supported by a transparent naming mechanism in the operating system. The operating system can also assist with replication transparency, leaving the task of fragmentation transparency to the distributed DBMS.

Full transparency is not a universally accepted objective. Gray argues that full transparency makes distributed data management difficult and claims that "applications coded with transparent access to geographically distributed databases have poor manageability, poor modularity, and poor message performance."⁵ He proposes a remote procedure call (RPC) mechanism between requester users and server DBMSs whereby users would direct their queries to a specific DBMS.

We agree that the management of distributed data is more difficult if transparent access is provided to users, and that the

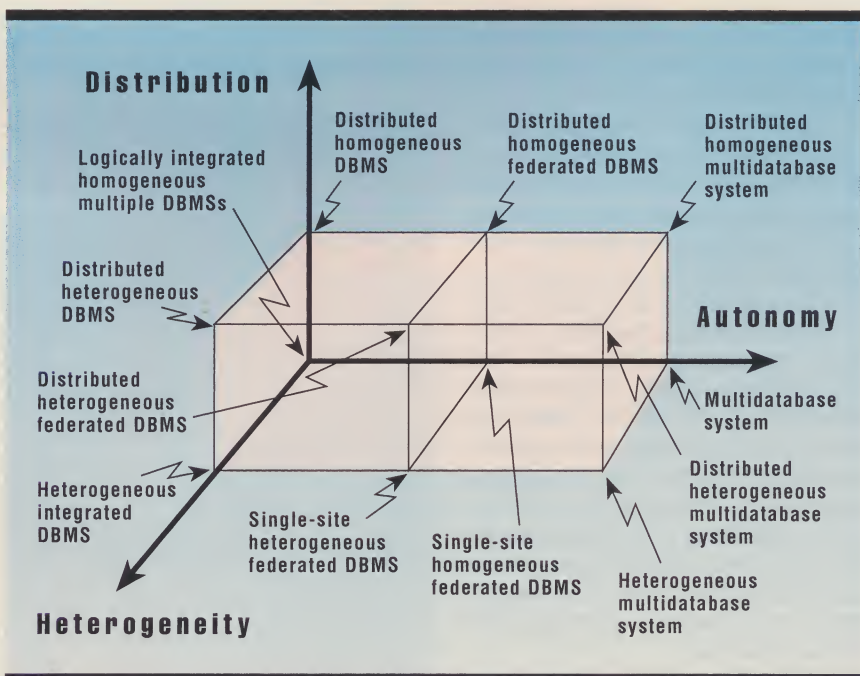


FIGURE 2. Implementation alternatives.

client/server architecture with RPC-based communications is the right approach. In fact, some commercial distributed DBMSs are organized in this fashion (Sybase, for example). However, the original goal of providing transparent access to distributed and replicated data should not be abandoned because of the difficulties. The issue is, what should take over the responsibility of managing distributed and replicated data—the user application or the distributed DBMS? In our opinion, it should be the distributed DBMS, whose components can be organized in a client/server fashion. The related technical problems are among the remaining research issues that must be addressed.

RELIABILITY

Distributed DBMSs should improve reliability, since they have replicated components and thereby eliminate single points of failure. The failure of a single site, or a communications link failure that makes one or more sites unreachable, is not enough to bring down the entire system. (It is well-known that link failures may cause network partitioning and are therefore more difficult to deal with. However, the topic is beyond the scope of this article.) In a distributed database, this failure means that some of the data may be unreachable, but with proper care, users can access other parts of the database. This proper care comes in the form of support for distributed transactions.

A transaction consists of a sequence of database operations, executed as an atomic action that transforms a consistent database state to another consistent database state, even when a number of such transactions are executed concurrently (sometimes called concurrency transparency), and even when failures occur (called failure atomicity). Therefore, a DBMS that provides full transaction support guarantees that concurrent execution of user transactions will not violate database consistency in the face of system failures as long as each transaction is correct—that is, obeys the integrity rules specified for the database.

Distributed transactions execute at multiple sites, where they

Distributed DBMSs should improve reliability

access the local database. With full support for distributed transactions, user applications can access a single logical image of the database and rely on the distributed DBMS to ensure that their requests will be executed correctly no matter what happens in the system. *Correctly* means that user applications need not be concerned with coordinating their accesses to individual local databases, nor need worry about the possibility of site or communications link failures during execution of their transactions. A link exists between distributed transactions and transparency since both involve distributed naming and directory management.

Providing transaction support requires the implementation of distributed concurrency control and distributed reliability protocols, which are significantly more complicated than their centralized counterparts. The typical distributed concurrency control algorithm is some variation of the well-known two-phase locking (2PL) protocol, depending on the placement of the lock tables and the assignment of lock management responsibilities. Distributed reliability

protocols consist of distributed commit protocols and recovery procedures. Commit protocols enforce atomicity of distributed transactions by ensuring that a given transaction has the same effect (commit or abort) at each site where it exists, whereas recovery protocols specify how global database consistency is to be restored following failures. In the distributed environment, the commit protocols are two-phase (2PC) protocols. In the first phase, an agreement is established among the various sites regarding the fate of a transaction. The agreed-upon action is taken in the second phase.

Data replication increases database availability; copies of the data stored at a failed or unreachable site exist at other operational sites. However, replica support requires the implementation of control protocols that enforce specified replica access semantics. The most straightforward semantic is one-copy equivalence, which can be enforced by the read one, write all (ROWA) protocol. In ROWA, a logical read operation on a replicated data item is converted to one physical read operation on any one of its copies, but a logical write operation is translated to physical writes on all copies. More complicated, less restrictive replica control protocols, based on deferring the writes on some copies, have been studied but are not implemented in any systems we know of.

Concurrency control and commit protocols are among the two most studied topics in distributed database research. Yet their implementation in commercial systems is not widespread. The performance implications of implementing distributed transactions, which are not fully understood, make them unpopular among vendors. Commercial systems provide varying degrees of distributed transaction support. Some (for example, Oracle Corp.'s Oracle) require users to have one database open at a given time, thereby eliminating the need for distributed transactions. Others (for example, Sybase Inc.'s Sybase SQL Server) implement the basic primitives necessary for the 2PC protocol but require the user applications to coordinate the commit actions. In other words, the distributed DBMS does not enforce ato-



micity of distributed transactions but provides the basic primitives by which user applications can enforce it. Other systems, however, implement the 2PC protocols fully (ASK/Ingres's Ingres and Tandem Computers' NonStop SQL, for example).

BETTER PERFORMANCE

The case for the distributed DBMS's superior performance is usually based on two points. First, a distributed DBMS fragments the conceptual database, enabling data to be stored in close proximity to its points of use. This feature, called data localization, has two potential advantages: Since each site handles only a portion of the database, contention for CPU and I/O services is not as severe as for centralized databases and, also, localization reduces remote access delays, which usually occur in wide area networks (for example, the minimum round-trip message propagation delay in satellite-based systems is approximately one second). Most distributed DBMSs are structured to gain maximum benefit from data localization. Full benefits of reduced contention and reduced communications overhead can be obtained only through proper fragmentation and distribution of the database.

The second point in favor of the distributed DBMS's performance advantage is that the inherent parallelism of distributed systems can be exploited for interquery and intraquery parallelism. Interquery parallelism results from the execution of multiple queries at the same time. Intraquery parallelism is achieved by breaking up a single query into a number of subqueries, each executed at a different site, accessing a different part of the distributed database.

If user access to the distributed database consisted only of querying (read-only access), provision of interquery and intraquery parallelism would imply that as much of the database as possible should be replicated. However, since most database accesses are not read-only, the mixing of read and update operations requires the implementation of elaborate concurrency control and commit protocols.

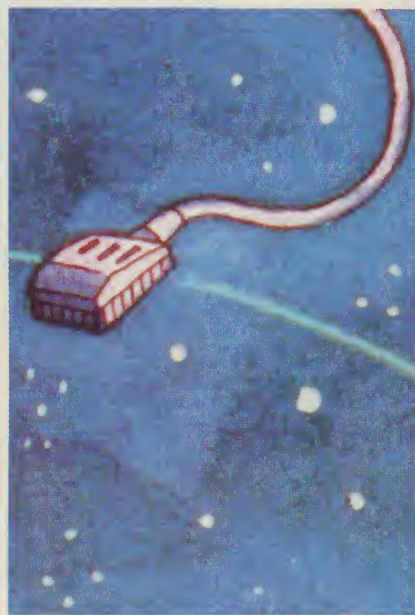
Today's commercial systems use two alternative execution models (other than the implementation

Most distributed DBMSs are local area network products

of full distributed transaction support) to improve performance. The first alternative is to have the database open only for queries (read-only access) during regular operating hours, while updates are batched. The database is then closed to query activity during off-hours, when the batched updates are run sequentially. This alternative is time-multiplexing between read activity and update activity.

The second alternative is based on multiplexing the database: Two copies of the database are maintained, one for ad hoc querying (called the query database) and the other for updates by application programs (called the production database). At regular intervals, the production database is copied to the query database. This alternative does not eliminate the need to implement concurrency control and reliability protocols for the production database since these functions are necessary to synchronize the write operations on the same data; however, it improves the performance of queries since they can be executed without transaction manipulation overhead.

The performance characteris-



tics of distributed database systems are not well understood. Not enough true distributed database applications are available to provide a sound basis for practical judgments. In addition, performance models are not sufficiently developed. The database community has developed a number of benchmarks to test the performance of transaction-processing applications, but it is not clear whether they can be used to measure the performance of distributed transaction management. The performance results of commercial DBMS products, even with respect to these benchmarks, generally are not openly published. NonStop SQL is one product for which performance figures, as well as the experimental setup used in obtaining them, have been published.

EASIER SYSTEM EXPANSION

In a distributed environment, accommodating increasing database sizes should be easier. Major system overhauls are seldom necessary; expansion can usually be handled by adding processing and storage power to the system. We call this *database size scaling*, as opposed to network scaling, which we will discuss later. It may not be possible to obtain a linear increase in power, since this increase also depends on the distribution overhead, but significant improvements are still possible.

Microprocessor and workstation technologies have played a role in improving economies. Many commercial distributed DBMSs operate on minicomputers and workstations to take advantage of their favorable price-performance characteristics. Moreover, most commercial distributed DBMSs operate within local area networks, for which workstation technology is most suitable. The emergence of distributed DBMSs that run on wide area networks may increase the importance of mainframes. On the other hand, future distributed DBMSs may support hierarchical organizations in which sites consist of computer clusters communicating over a local area network, with a high-speed backbone wide area network connecting the clusters.

Another economic factor is

the trade-off between data communications and telecommunications costs. In the previous section, we argued that data localization improves performance by reducing delays. It also reduces costs. Consider an application (such as inventory control) that needs to run at several locations. If this application accesses the database frequently, distributing the data and processing it locally may be more economical than executing the application at various sites and making remote accesses to a central database stored at another site. In other words, the cost of distributing data and shipping some of it periodically from one site to the other to execute distributed queries may be lower than the telecommunications cost of frequently accessing a remote database. This part of the economics argument is still speculative. As we indicated before, most distributed DBMSs are local area network products, and how they can be extended to operate in wide area networks is a topic of discussion and controversy.

MORE TO COME

Next month, we will discuss the unsolved problems of distributed DBMSs, and the new issues associated with this changing technology. ■■■

©1991 IEEE. Reprinted with permission from Computer (Vol. 24, No. 8, pps. 68-78, August 1991).

The authors would like to thank Abdel-salam Heddaya of Boston University, who not only reviewed the entire article and provided many comments, but also helped with the discussion of replication, which is based on a draft he wrote. Thanks to Alex Biliris of Boston University, and Michael Brodie, Alex Buchmann, Dimitrios Georgakopoulos, and Frank Manola, all of GTE Laboratories, who also read the entire manuscript and provided many suggestions regarding content and presentation that improved the article significantly.

M. Tamer Özsu's research was partially supported by the Natural Sciences and Engineering Research Council of Canada under operating grant OGP-0951.

REFERENCES

1. Stonebraker, M. *Readings in Database Systems*, Morgan Kaufmann, San Mateo, California, 1988, p. 189.
2. Stonebraker, M. "Future Trends in Database Systems," IEEE Trans. Knowledge and Data Eng., 1(1): 33-44, March 1989.
3. Garcia-Molina, H. and B. Lindsay, "Research Directions for Distributed Data-

bases," IEEE Q. Bull. Database Eng., 13(4): 12-17, December 1990.

4. Özsu, M.T., and P. Valduriez, *Principles of Distributed Database Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

5. Gray, J. "Transparency in Its Place—The Case Against Transparent Access to Geographically Distributed Data," Tech. Report TR89.1, Tandem Computers, Cupertino, California, 1989.

M. Tamer Özsu is an associate professor in the Department of Computing Science at the University of Alberta (Edmonton, Canada), where he leads a research group that investigates distributed databases, object-oriented databases, and database operating system design issues. He is the author or coauthor of 12 books as well as a number of technical papers dealing with database technology.

Patrick Valduriez is a director of research at INRIA, the national research center for computer science in France. He is currently heading a project on advanced database technology, including rule-based and object-oriented databases and parallelism. He is the author or coauthor of over 50 technical papers and several books on database systems.

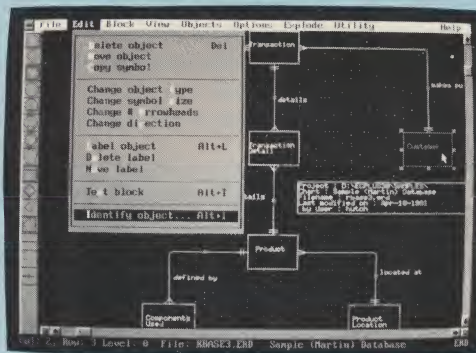
To contact the authors, write to M. Tamer Özsu, Dept. of Computing Science, University of Alberta, Edmonton, Canada T6G 2H1. His e-mail address is ozsu@cs.ualberta.ca.

EasyCASE™ Plus

COMING SOON...
Schema Generation
for xBASE & SQL!

The affordable approach to software engineering... only \$495

Finally, there's a CASE tool that won't get in the way of your creativity... A tool that makes structured analysis, structured design and data modeling as easy as working with any other tool on your PC - EasyCASE Plus! Using EasyCASE Plus' new, easy to use graphical user interface (GUI), you'll be creating and editing charts, linking them, and building your data dictionary in no time. As well as being easy to use and easy to learn, EasyCASE Plus is easy on your budget! Ask any user. They'll tell you it's the best buy for your PC based CASE tool needs. Discover why over 4,000 software professionals use EasyCASE Plus and how you can join them!



"EasyCASE Plus is a well designed, low priced tool that is easy to learn and provides excellent diagramming capabilities... EasyCASE Plus is an excellent investment."

COMPUTER
LANGUAGE
PRODUCTIVITY
AWARD
1990

Features:

- IBM SAA/CUA compliant graphical user interface (GUI)
- Extensive diagram editing features
- Integrated dBASE III compatible data dictionary
- Integrated dictionary manager, reports manager, process editor
- Hierarchical chart linking & process decomposition
- Record and element definitions
- Extensive printer, plotter and desktop publishing support
- Data dictionary import, export, and merge
- On-line help
- Comprehensive documentation with tutorial
- Access to your database, word processor, DOS, etc.
- Integrated diagram analysis (optional)

Requirements:

Runs on: IBM PC or PS/2 (AT recommended), DOS 3.1 or higher, EGA/VGA color, mouse, 640 K RAM (500 K free), 1 MB EMS recommended, math co-processor supported. Printers/Plotters Supported: Epson FX & LQ, IBM Graphics & Proprinter X24, HP QuietJet, DeskJet, & LaserJet, HP Plotters, PostScript.

EasyCASE Professional \$649
(includes integrated DFD level balancing and data dictionary/diagram analysis)

Methods:

- Yourdon/DeMarco
- Gane & Sarson
- Ward-Mellor/Hatley
- Yourdon/Constantine
- Martin
- Chen, Bachman

Diagram Types:

- Data Flow Diagrams (DFDs)
- Structure Charts
- State Transition Diagrams
- Entity Relationship (ERDs)
- Data Model Diagrams
- Transformation Schema (real-time DFDs)

Evergreen
CASE
Tools

16650 NE 79th Street
Suite 200
Redmond, WA 98052
USA
FAX: (206) 883-7676

Call today for a brochure!
Tel: (206) 881-5149

©1991 by Evergreen CASE Tools, Inc., All Rights Reserved. All trademarks are the property of their respective companies.

CIRCLE 25 ON READER SERVICE CARD

MARCH 1992

INTERNATIONAL DB2 USERS GROUP

4th Annual International Conference

"Target the Future — Managing Your DB2 Environment"

May 10-14, 1992

New York Hilton & Towers

IDUG Conference Brings Unduplicated Education and Networking Experiences to All Levels of DB2 Users

Beginning May 10 through May 14, 1992, the International DB2 Users Group (IDUG) builds on the experience and strengths of its three previous conferences to bring knowledgeable relational technology industry sources and technical education opportunities to the New York Hilton & Towers in New York City. The event is IDUG's 4th Annual International Conference, "Target the Future — Managing Your DB2 Environment." The value is an unprecedented selection of quality training sessions critical to productive and effective job performance.



Bringing together more than 1,800 DB2 users and suppliers from around the world, this major industry event features expert speakers and technical sessions geared to nine key levels of relational technology application and experience, including a new focus on Executive Development. They are:

- | | |
|--|-----------------------------------|
| • Systems Programming | • Distributed Technology |
| • New DB2 Users | • Advanced DB2 Users |
| • Database Administration | • Performance |
| • Application Development/
Data Modeling/CASE | • Decision Support/
Management |

Other highlights are the new "Star Track" featuring speakers considered to be the best of the industry; an "Encore Track" offering second opportunities to hear favorite speakers; and, for the first time, "Exhibit Only" passes.

What Makes IDUG Conferences Worth the Investment?

IDUG is an independent, non-profit, volunteer-driven organization consisting of user, associate and vendor members. IDUG conferences stress objective, fresh presentations geared strictly to users, while vendor and IBM participation bring the conference full circle. Meetings of special interest groups, highly informative panel presentations, and an impressive display of new products and technology provide conference

participants with unmatched take-home value. Peer networking opportunities — among a strong international presence — are limitless. Just as important, the cost to attend is minimal; \$795 (before March 20) purchases a full conference registration, including meals.

This year's keynote speakers include:

Jeff Tash, founder and president of Database Decisions, presenting "DB2 as a Foundation for IBM's Frameworks";

Earl Wheeler, senior vice president and general manager of Programming Systems for IBM, discussing IBM's newest composite application, "The Information Warehouse Framework";

Colin White, founder and president of Database Associates International, presenting new information on "DB2 Interoperability"; and

Gabrielle Wiorkowski, founder and president of Gabrielle & Associates, presenting "Index Design, Joins, and Subselect Performance."



INTERNATIONAL
DB2 USERS GROUP

DPD

Return this coupon to the International DB2 Users Group, 401 North Michigan Avenue, Chicago, Illinois USA, 60611-4267 (FAX 312/321-6869), to receive information on user membership and IDUG's 4th Annual International Conference. For more information, call IDUG at (USA) 312/527-6777.

Name _____

Title _____ Company _____

Business Address _____

City _____ State/Province _____

Country _____ ZIP/Postal Code _____

Business Phone _____ FAX _____

☐ If you need vendor-related information on membership and the many conference participation opportunities, mark this box.

Windows Client/Server Workshop

Test Driving Tools for the New Application Development Environments

Windows, Presentation Manager, Apple, Motif and OpenLook

Boston ■ April 28-30, 1992

*How do you build client/server applications that run in a windows environment?
Are you in a quandary, unsure of what to do, what to buy, or who to buy from?*

To help you compare and contrast the full power of various windows products' features and benefits, each vendor at The Windows Client/Server Workshop has **75 minutes to walk you through an entire application development effort, from soup-to-nuts.**

Windows tools must be seen in action because these products defy simple comparisons. Many are radically different, particularly in their approach to application development, that's why **live demos are mandatory.**

To assist you in your windows tools selection process, The Windows Client/Server Workshop ends each day with hospitality suites where you can "roll-up-your-sleeves" and "test drive" the previously demo'd products.

A partial listing of vendors who will showcase their windows tools:

- Cognos
- Cooperative Solutions
- Matesys Corporation
- Micro Data Base Systems, Inc.*
- ONTOS, Inc.*
- ParcPlace Systems, Inc.
- Powersoft*
- Revelation Technologies, Inc.*

Shouldn't You Be Participating Too?
Call Kathleen Spencer at
(508) 470-3870 for More
Information.

* Conference Co-Sponsor

Added Educational Value

The Windows Client/Server Workshop also offers you the chance to hear leading consultants and industry gurus discuss the major issues surrounding windows client/server technology.

Speaker Presentations

- **Frameworks for Client/Server Computing** by Conference Chairman Jeff Tash, Database Decisions
- **LAN Survival Guide** by Larry DeBoever, Tucker/DeBoever Technologies
- **Integrating Client/Server and CASE** by Pieter Mimno, Technology Insight, Inc.
- **Product Comparisons: Strengths, Weaknesses and Commentary** by Conference Chairman Jeff Tash, Database Decisions

Technical Presentations

- **Effective GUI Design** by Christine Comaford, Corporate Computing, Inc.
- **Dynamic Data Exchange (DDE) and Object Linking & Embedding (OLE)** by Kim Crouse, Synaptix
- **Beyond CUA, How to Set Successful GUI Design Standards** by Christine Comaford, Corporate Computing, Inc.
- **Windows Networking: Using IPCs & RPCs** by Greg Denenfeld, Denenfeld Systems Design

Hot New Products and Companies to Watch

This panel discussion, moderated by Fred Langa, *Windows Magazine*, brings together a who's who of distinguished computer industry press members who will share their opinions and ideas about the latest and greatest new products and companies on the windows scene. Panel includes representatives from *PC Week*, *Computerworld*, *Info World* and *Byte*.

Call today to register or receive more information on this exciting new workshop (508) 470-3880.

Sponsored by



Digital Consulting, Inc.



0A32DF

CIRCLE 26 ON READER SERVICE CARD

Primary keys—the building blocks of a stable database—are often compromised to fit real-world situations. Here's a technique to provide controlled flexibility

How and When to Use Surrogate Keys

SURROGATE KEYS (artificial, meaningless arbitrary identifiers) are often proposed as alternatives when problems with primary keys occur. But what factors must data analysts and database designers take into consideration to determine if surrogate keys are best suited to take the role of primary identifier?

First, does an attribute or set of attributes that has all the required properties of a primary key exist? A primary key should be unique to differentiate one occurrence of an entity from another. A primary key should also be explicit, in that a value exists for every entity occurrence known upon its creation; definite, so its initial value is not null; consistent, so each occurrence is associated with only one value of its identifier and the value the same wherever that occurrence is referenced; and stable, so its value never changes or becomes null as long as that occurrence exists. A primary key should also be factless so it identifies but does not describe the entity occurrence. A factless identifier increases the likelihood that the primary key will be unique, explicit, definite, consistent, and stable.

In addition to these fundamental properties of primary keys, the attributes are ideally accessible and understandable to users and conform to any DBMS or installation-specific limitations. And whenever possible, a primary key should also be controllable so a suitable identifier can be assigned.

PRIMARY KEY PROBLEMS

Given this impressive list of qualifications for attributes aspiring to be primary keys, no wonder problems occur in the real world. The first problem is a primary key whose value is subject to change, could become null, or is comprised

of attributes that could become nonkey, which is known as a volatile primary key. If the primary key can change, the uniqueness of its value may be compromised. In addition, if the primary key can become null, the properties of consistency and explicitness are also in jeopardy since a null does not signify a valid value.

For primary key attributes that become nonkey, the stability, consistency, and explicitness of the primary key is affected. The data administrator must also be concerned with updating the logical data model and data dictionary entries to reflect the removal of this attribute from the primary key.

A volatile primary key can create problems with maintaining referential integrity as well as performing joins and other operations (such as attribute concatenation), which are impacted by nulls. And for ordering data, nulls will either sort to the end or beginning of the output list, depending on the chosen relational DBMS.

The second problem concerns the concatenation of multiple attributes into one denormalized primary key attribute. Because multiple attributes are combined into one, the business meaning of data is not clear, which could hamper user accessibility. If any part of the primary key attribute has the potential to change or become null, it is more difficult to deal with than if the attributes had remained separate. A denormalized primary key also presents problems with joins, range-checking, sorting, ordering,

grouping, index definition, application performance, referential integrity, and domain enforcement. Because the attributes must always be treated as a unit, some operations are precluded and others accomplished only through the use of inefficient operators, such as substrings.

The third problem is an all-purpose primary key. In this case, either one attribute or another can function as the primary identifier because values never occur simultaneously in both. Both attributes are layered on top of each other to create a primary key that does not have a null component. The all-purpose primary key is created by taking the most forgiving data type (usually character) and longest length of the two attributes, calling it some combination of their names, and populating it with whichever value is present in each entity occurrence.

An all-purpose primary key is, in effect, a form of a denormalized primary key since the primary key attribute is really comprised of two attributes. Instead of denormalization through the concatenation of attributes, the attributes are superimposed upon one another. This approach could compromise primary key uniqueness if the range of values for the two attributes overlaps, and if it is difficult for the user to understand the attribute's business meaning since it means different things in different circumstances. Overlapping logical domains can also complicate range-checking and ordering or group-

ing if data keyed by one attribute is to be separated from data keyed by the second attribute.

A primary key that is not universally agreed upon poses the last problem. This situation is called a nonuniversal or parochial primary key and occurs when the entity life-cycle phases fall in different functional organizations. For example, the planning department may identify a product by a different set of attributes or at a different level than the manufacturing department, and so on through the warehousing, marketing, and sales departments. Even if an attribute of the same name is used, such as product number, it could have different logical and physical domains.

For this situation to occur, data models for each functional area would have had to be developed independently. Enforcing uniqueness across functional areas is an impossibility when the same primary key is not used by all areas. User accessibility and understandability of the data across departments is similarly impossible, prohibiting any meaningful cross-departmental or historical tracking. Nor can the primary key be considered stable when it can take on different department-specific physical representations, values, and meanings.

CONSIDERATIONS

While another attribute or group of attributes that better fulfills the properties of a primary key (known as an alternate key) may exist, a surrogate key should always be considered. Because of the rapidly changing nature of business, a possibility always exists that any identifier that is not factless could become volatile. But before you decide to use a surrogate key, you must weigh a number of other factors.

The length of the primary key should be considered. Long primary keys have implications for DASD consumption to store the indexes supporting the primary key and its occurrence in other tables as a foreign key. A long primary key also impacts the processing time required for such utilities as load, backup, and recovery, and such operations as joins, scans, inserts, and deletes. A surrogate key has the advantage of being relatively short in comparison.

The length of the primary key should be considered

If the primary key is a composite or multicolumn key, all columns of the primary key must be specified to retrieve a row or perform a join. Creating views to define selection or join criteria or create synonyms for the concatenation of the columns can alleviate this problem. All columns are also needed to maintain referential integrity, which could prove onerous. A multicolumn primary key also may be very long, with all the attendant concerns thereof. A surrogate key is almost always a single attribute, which simplifies referential integrity and the coding of SQL statements.

Index limitations posed by the chosen DBMS must also be considered. An index limitation on an index's length or the number of columns it could reference could preclude the use of an index to enforce uniqueness. In this case, you can enforce uniqueness through application logic, so a surrogate key may provide a good alternative.

The existence of any of these factors will tip the scales in favor of using surrogate keys. The following considerations may weight the decision in the other direction.

Installation-specific limitations can play a role in the decision to use surrogate keys, especially when coupled with a requirement for user access by the original primary key rather than the surrogate key. The DBA may place a limit on the number of indexes that can be defined on a table. As more indexes are defined on a table, the amount of required DASD increases, as does the probability of locking problems; the processing time required for inserts, updates, reorganizations, recovery, backup, load, and statistics gathering; the object administration to create the indexes; the probability of reaching the limit on the number of open datasets; and the complexity of table or partition recovery. If users require access by the original primary key

and an installation limitation on the number of indexes allowed is present, using a surrogate key may not be feasible.

The primary determinant in using a surrogate key is which organization owns the business information and assigns the primary key. If the primary key is assigned externally, a surrogate key probably will not be the appropriate choice due to the inability to identify the same entity occurrence across organizational boundaries consistently. If the primary key assignment is internally controlled, the decision to use surrogate keys is entirely up to the organization's database designers and is based on a thorough analysis of the business information requirements.

However, an externally assigned primary key is not controllable, probably not factless, and more prone to volatility and other primary key problems. If this is the case, then a surrogate key may be used internally but the original primary key attributes must be maintained for providing cross-organizational tracking.

Another exception to using surrogate keys for externally controlled data is when two self-contained systems with different primary keys are integrated and surrogate keys are already being used by the target system. You should create a cross-reference of the original primary key to the new surrogate primary key to provide a bridge between systems as long as required. If the primary key assignment is internally controlled, the decision to use surrogate keys is entirely up to the organization's database designers.

If a surrogate key is assigned as the primary key, the values must be made known to the users immediately. The original primary key attributes should be included as columns in the table and an index should be defined on them if user access warrants it and installation limitations allow.

Once you make the decision to use surrogate keys, implications for different types of indexes exist. Using a random assignment algorithm for a surrogate key risks the minute possibility of duplicate values. This risk can be controlled through application logic to gen-

erate another value should a duplicate be encountered. A surrogate key functioning as a primary key is not a good basis for a clustered index since the values have no inherent business meaning. While a unique index can be defined on the surrogate key, a clustering index on the original primary key or another attribute may better support access requirements. In tables where the surrogate key functions as a foreign key, the definition of a clustering index on the surrogate key may be appropriate to access the data as required. An example of such a clustering index would be on the surrogate key customer number in a customer invoice table.

When partitioning is required, serial assignment of a surrogate will result in a skewed distribution of rows across partitions. The first partition would be completely filled before any rows are inserted into the second partition. Because of the amount of activity occurring on one partition, locking problems may ensue.

An alternative to randomly assigned surrogate keys for resolving this problem is a modified serial assignment algorithm. Instead of vertically filling each partition before going on to the next, the surrogate key values can be assigned to insert one row into each partition before inserting a second. This horizontal assignment ensures a uniform distribution of rows across partitions and eliminates potential locking problems.

ALTERNATIVES

If primary key problems prohibit the use of the original primary key and if the surrogate key is not a viable alternative, other options exist. A timestamp can be used as the primary key or a primary key component. Except in the case of batched arrival, uniqueness would be ensured. A timestamp could also be meaningful if arrival sequence has importance to the entity in question.

A timestamp, however, can be quite long (26 bytes in DB2), which dramatically increases the size of the primary key and raises the issue of all the DASD, performance, and index limitation considerations included in the discussion on sur-

A final alternative to surrogate keys is the substitute key

rogate keys. In addition, the timestamp represents adding another column to the primary key—so the multicolumn primary key considerations apply as well. And it does not resolve the problems associated with externally originating data since the only controllable part of the primary key is the timestamp. Also, clustering and partitioning indexes should not be defined on a primary key that consists of a timestamp alone because locking problems could ensue with a large volume of inserts.

While a surrogate key may not work as the primary key, it may be suitable as a primary key component. If the primary key attributes do not ensure uniqueness, adding a surrogate key in the form of a sequential counter will. This addition is commonly made to identify invoice or order line items.

Adding a sequential counter does not increase the primary key's overall length by much, but it may raise the issues of DASD consumption, performance degradation, and installation-specific index limitations I discussed earlier. If the data originates in an external organization, the only controllable part of the primary key is the sequential counter.

A final alternative to surrogate keys is the substitute key. The value of the primary key or any of its components is abbreviated or encoded to reduce its length. All substitute values are predetermined to represent a specific business meaning prior to their assignment as a primary key. This approach is unlike the surrogate key, which has no inherent business meaning and whose value is determined at time of assignment. The abbreviations for airline carriers and airports are examples of substitute keys.

Substitute keys can substantially reduce the primary key length, so many of the length-related considerations are no longer an issue. However, controllability will still be an issue with externally origi-

nating data. Only the assignment of the substitute key is controllable. Using substitute keys with external data may not be possible if the external data is volatile. The designation of the valid substitute keys cannot be made on the fly. It must be made in advance of the assignment of the substitute key to entity occurrences.

An alternative for addressing denormalized and all-purpose primary keys is the controlled introduction of data redundancy. If some applications use denormalized or all-purpose keys and others do not, the original application with primary key problems can introduce redundancy by including the component attributes as nonkey data elements in the table using the group or layered attribute as the primary key. Additional indexes to support joins and access requirements would have to be defined, installation limitations permitting. This approach would avoid the complicated application programming required by using substringing and concatenation.

Doing nothing is also an option. To an extent, the programming complexities can be mitigated through the creation of views for joins and selections requiring the use of string operators. However, the performance problems and inability to perform operations such as joins and attribute concatenation will still exist.

TRICKS OF THE TRADE

Surrogate keys are not the "silver bullet" for primary key problems. But under the right set of circumstances, surrogate keys can meet the requirements of a primary identifier without creating additional problems for users and database designers. The real trick is to know when they are appropriate and when some other alternative would be a better fit. ■■■

REFERENCES

Whitener, T. "Primary Identifiers: The Basics of Database Stability," *Database Programming and Design*, 2(1): 42-49, January 1989.

Deborah L. Brooks is a data specialist at MCI Telecommunications Corp. in Arlington, Virginia. She is the author of articles on data prototyping and data administration and has spoken at the International DB2 User's Group (IDUG) on surrogate keys.

The concluding part of our DBMS selection criteria will help you decide if a product can handle systems, architectural, and operational issues

Navigating the DBMS Labyrinth

SYSTEM INTERNALS and architectural issues cover items especially critical to those who must design an application architecture or work with the DBMS engine's internal workings. Some of the categories covered in the checklist over the past two months have addressed very specific areas of functionality. However, this month, I'll address the product's major design components and then focus within those components. Since these areas compose the heart of a DBMS system, the design decisions a vendor makes often affect the direction and composition of a product and product family for many years to come.

SECURITY

In the broadest sense, you can think of security as the ability to control access to a database and its data. When examined closely, security can either be present or not present in many areas, and each one can take on multitudes of implementations. Almost without exception, the concept of security is of utmost importance to most applications. Therefore, you must understand the levels of security a product supports and how they are supported. To understand this category, examine these areas:

- ☐ Does the DBMS depend on operating system security only or does it use its own security system on top of the operating system?
- ☐ Does the DBMS support the concept of DBMS defined users? If so, what level of user and user group administration is necessary? How are remote users incorporated into the security system?

☐ How is database and table ownership determined?

☐ Does the database provide the ability to control entrance into the database? Can entrance be controlled for a single user, group of users, or all users?

☐ Can you control who creates tables and indexes and alters system catalogs? Are grantable permissions supported on the data-definition language?

☐ What is the create, retrieve, update, delete (CRUD) level security for tables? Are grantable privileges available to create, retrieve, update, and delete records from tables? Are they supported at the single user, group, or public level?

☐ Can you establish permissions within a table; for example, permission on a row, column, field, and so on?

☐ Are permissions flexible enough to support time of day and day of week sensitivity (for example, providing a user with access to data from 9 A.M. to 5 P.M., Monday through Friday only)?

☐ Do permissions extend to views? Are they separately definable?

☐ Does security extend to stored procedures and rules?

☐ Is the concept of a private database, which is accessible only by its creator, supported?

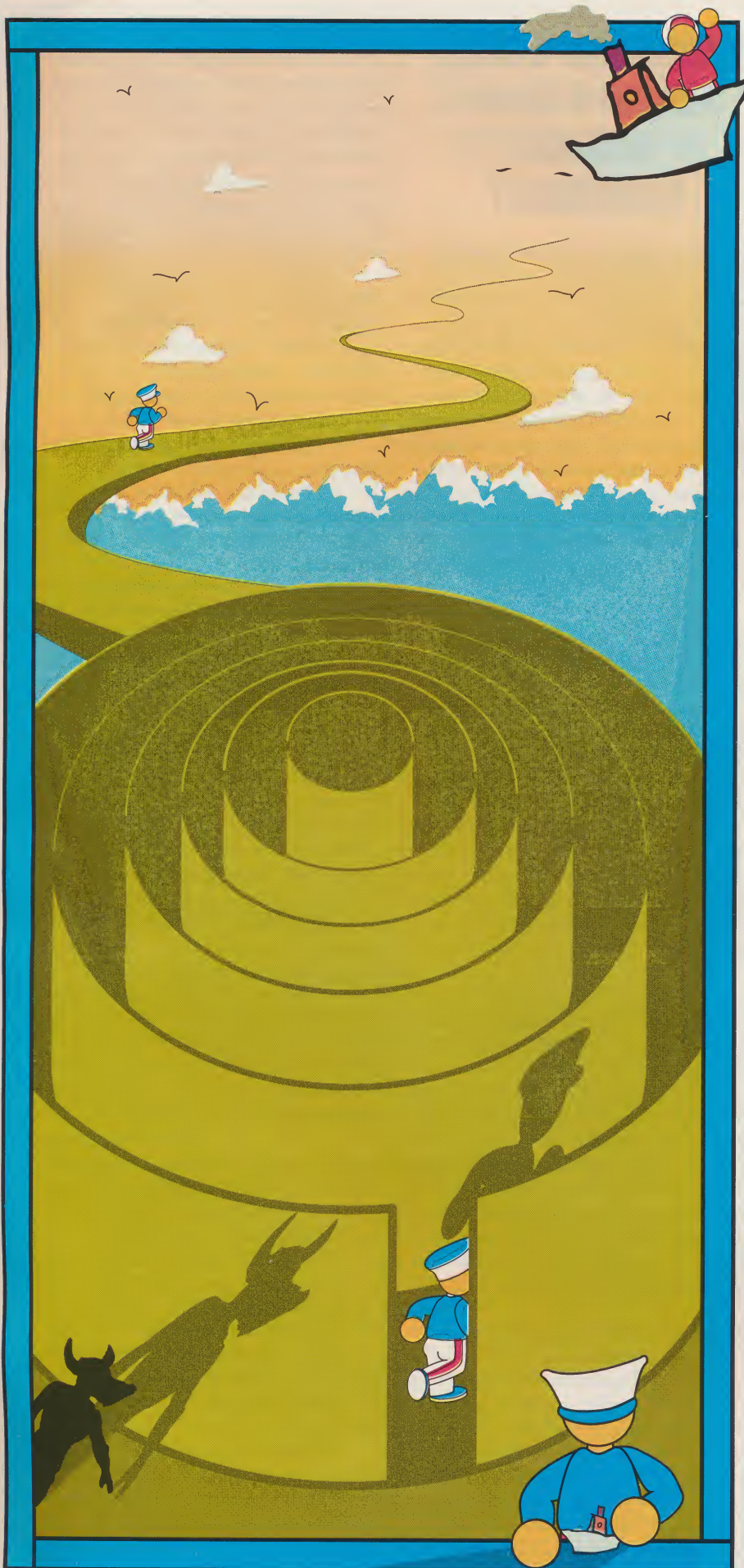
☐ What type of security is available on database journal and log files?

☐ How is transaction data, which is sent over a network, secured?

Obviously, different products implement many different security features. Some of these features may be useless to some applications and indispensable to others. When evaluating a DBMS, you must determine your security needs before investigating a product's functionality in this area. Failure to do so will often result in a key requirement being overlooked until it is too late to implement it during development.

CONCURRENCY CONTROL

Concurrency control is the DBMS's ability to manage the isolation and locking of all or parts of a database. This function occurs in a multiuser environment when multiple users attempt to update a database



at the same time. In these cases, the DBMS must assure that the database remains in a consistent state throughout all operations. This aspect of a DBMS may seem too low level to be very important to customers; however, a DBMS will often provide various levels of concurrency management that provide customers with numerous design and development options. Some areas to focus on include:

☐ How does the DBMS control database locking?

☐ What types of locks are supported (shared, exclusive, intent to update, and/or demand)?

☐ Is lock escalation supported? (For example, if you lock a great number of individual data pages in a table, it is often more efficient to lock the entire table in lieu of pages.) Is this feature automatic or can it be controlled?

☐ Does the DBMS support promotable locks? (For example, if you already hold a shared lock on a record, can it be promoted to an exclusive lock without releasing the shared lock?)

☐ Are time-outs for locks available? How are deadlocks detected and handled?

☐ What is the default locking?

☐ How does locking work in a distributed environment (for example, locking among processors)?

☐ What level of locking granularity (page, row, table, database, and so on) is supported?

☐ Can you tune the locking system?

The amount of flexibility in the concurrency control system often relates directly to the level of performance that can be attained. Therefore, it is an area you should examine in some detail, particularly if you are familiar with the nature of an application's transactions and their interaction.

QUERY OPTIMIZER

A query optimizer is perhaps the one aspect of a relational DBMS that contributes most to the system's performance. You can think of the optimizer as built-in intelligence that determines the best access path to the requested data. This process may sound rather simple but, on the contrary, is very complex because determining the best path to data is not only diffi-

cult but may mean different things to different people. An intelligent query optimizer can often remove a burden from a developer's shoulders and compensate for the lack of programmer expertise. Never ignore the query optimizer in a DBMS evaluation. Some specific features to analyze include:

☐ How does the optimizer determine the best path to the data? Does it use statistics the database keeps about the data, determine the least-cost method, require the programmer to specify the access path by the query's order, or determine an access path based on predetermined rules?

☐ Can a DBA specify data statistics or update the system's statistics? Does the optimizer automatically update statistics, or can a user access and manually update these statistics? Can users request an update, or must they update manually?

☐ Can the optimizer account for differences in data distribution (skewed data patterns, key uniformity, and so on) in a table?

☐ What does the optimizer take into account (for example, numbers of rows in tables, indexes, unique key values, estimated CPU and I/O, estimated data transfer to a networked environment) to determine the access path?

☐ Can the optimizer estimate query costs? How does it define them?

☐ Is there a query explain facility that supplies the optimizer's access choices prior to the query's execution?

☐ Can you override an optimizer's access choice?

In general, the more intelligent the optimizer, the less knowledgeable the user must be. Of course, intelligence comes at a price, which, in many cases, is the system overhead for determining the access path. For simple queries, it may take more time to determine an access path than to access the data! Of course, a good optimizer more than makes up for this trade-off in the complex query case. Usually, an item such as a query explain facility is an invaluable tool to any development shop.

OBJECT MANAGEMENT

Many think that, after relational, object-oriented database represents

Does the DBMS support user-defined functions?

the next major database paradigm. However, it remains to be seen whether an entire new breed of products will displace today's popular choices. Nevertheless, to remain state-of-the-art, today's market leaders will have to incorporate some degree of object-oriented features and knowledge management into their products. In some cases, a vendor may have already expanded a product along these lines; in others, plans may be on the drawing board. To identify these differences, you should examine the following:

☐ Can the concept of inheritance be implemented among objects? Are object types and subtypes allowed?

☐ Does the DBMS support user-defined functions? Are they enforced at the DBMS level?

☐ Does the DBMS support recursive and nested functions?

☐ Can large nontraditional objects (image, voice pattern, and so on) be stored in a database file? If so, how are they manipulated?

☐ Are standard DBMS security features in use for these large objects?

☐ What storage medium is used to house large binary data?

☐ Can the DBMS tool sets display such images?

Although these features are somewhat advanced at present, they will become even more important in the future. If you're currently considering a DBMS product, you should understand what the vendor's plans are in these areas to avoid being behind the technological times in the future.

DISTRIBUTED PROCESSING

From a database perspective, you can think of distributed processing as the ability to make a request from data on a processor that is remote to the requestor. A subset of distributed processing, distributed database expands the concept by permitting data residing on separate processors (or at least in sepa-

rate databases) to be viewed and accessed as if it were a single database. These architectural issues are among the most critical to many customers today, and it is therefore crucial to understand the implementation of distributed features by the major vendors. The following questions raise some important considerations:

☐ Does the product provide homogeneous read capability (the ability to read remotely from the same DBMS product in a multiprocessor, networked environment)? Must both DBMS products be at the same release level?

☐ What is the connecting software and underlying networking?

☐ Does the product support downloads or "snapshots" of remote data only or are multisite joins supported?

☐ How does the DBMS determine where the data resides? In the case of a multisite table join, how does it determine the optimal location (the most efficient manner in terms of processing and network traffic) to perform the join?

☐ Does the product maintain a "global data dictionary" or global catalogs? Is it stored on one or many sites? (If the product is stored on one site, it runs the risk of a single point of failure that can stall all distributed processing. If it is stored on multiple sites, the product must keep all the sites in sync with one another as the database changes.)

☐ How can a distributed database be partitioned? (Some possibilities include by table, by columns within tables, by keys within tables and so on [for example, horizontal versus vertical partitioning].)

☐ Does the product support remote access for heterogeneous products? If so, which products does it interface with, and what are the software and networking components? What limitations does the product impose on access? Can it interface to nonrelational products?

☐ Does the product support homogeneous write capability (the ability to update a remote database in a multiprocessor, networked environment)?

☐ Does it permit single-site updates only or does it support a two-phased commit protocol that permits simultaneous multisite updates?

☐ Does the product support heterogeneous write (remote updates of foreign DBMS products)? If so, can it act as the coordinator of such a transaction? Will it function as a participant to another coordinator?

☐ How are deadlocks handled in the distributed case (for homogeneous and heterogeneous products)?

☐ Can the product participate in two-phased commits driven by a transaction processing monitor?

☐ Which processors, networks, gateways, and product versions are supported by this product in its distributed architecture?

☐ Does the product support the ISO RDA standard?

The number of questions on this list gives you a fair impression of the complexities raised by distributed computing and distributed database. Before approaching vendors on this issue, you must understand your business computing needs. In many cases, a distributed solution is not the proper business solution. Nevertheless, with more and more powerful desktop resources, distributed processing should always be a consideration. Therefore, you must be aware of the vast array of implementations and their implications.

NETWORK COMPATIBILITY

Tied very closely to distributed processing are underlying network compatibility issues. DBMS vendors must typically make many choices as to their preferred architectural offerings in terms of processors and operating systems supported and the various networking components that tie the systems together. Needless to say, it is mandatory that the network architectures entrenched in a corporation are among the supported vendor offerings. Some key network issues may include:

☐ Can the product run in a Datakit environment? STARLAN? FDDI? XNA? StarGroup?

☐ Does it provide connectivity over a TCP-IP gateway? TCP-IP over X.25? LU6.2?

☐ Can it provide connectivity directly over X.25?

☐ Can it run over Ethernet? OS/2 LAN Manager? LAN Manager/X? How does it connect (Net-

Sync replication is a specialized architectural need

Bios, Named Pipes, and so on)?

☐ What is the vendor's commitment to OSI standards?

In this area, you should be familiar with the key networking requirements for your particular environment. For example, if you are in the market for a LAN Server DBMS, you may not care much about the interface from UNIX to MVS. However, you would need to evaluate the implementation of the networking from client to server over the LAN.

SERVER COMPATIBILITY

If you are interested in client/server computing and, therefore, positioning the database as a server, it becomes important to understand how well the DBMS can function in the server role. This role is one in which the database resides on one processor and is available to serve the data needs of many remote clients. To perform this role effectively, the vendor must address the following:

☐ How effective is its memory utilization on the client and server processors?

☐ What is the interface to the network? Does the product attempt to minimize traffic?

☐ Does the DBMS support stored procedures and triggers? (By grouping requests into callable procedures and letting them trigger other procedures, the server can perform much work independent of the client, thus reducing network traffic and overhead.)

☐ Can the product interoperate with a large variety of clients? (This issue is especially pertinent for a LAN server that may need to work with third-party, shrink-wrapped software.)

☐ How accessible is the server? Which platforms can easily get to the data stored on the server, using which mechanisms?

Although many traditional DBMS issues are also issues when considering a DBMS as a server,

the focus may shift to interoperability and accessibility and how well the product can function in the server role. A product that performs exceptionally well in a host-based environment may fail as a server if specific "server-type" features are not supported.

SYNC REPLICATION

Sync replication refers to the degree of fault tolerance in which an application can maintain an up-to-the-second, stand-by copy of a database. In the case of failure or corruption of the primary database, the stand-by is simply brought in and the application continues processing as if nothing happened. Of course, this functionality is dependent on hardware and DBMS support. From the DBMS standpoint, the issues include:

☐ Does the product support database shadowing or real-time replication?

☐ Is a dual-copy, mirror image maintained, or does the product rely totally on hardware features, such as disk mirroring?

☐ Can a dual copy be maintained across multiple processors and networks or must they reside on the same machine?

☐ How is the primary copy brought back in sync with the stand-by copy after the primary has been brought back online?

☐ At what granularity (full database, tables, indexes, and so on) is sync replication supported?

Sync replication is a specialized architectural need. Many applications do not need this level of resilience; however, for those that need it, the fault tolerance capabilities of the product become key.

DEFERRED COPY MANAGEMENT

Deferred copy management is a milder version of sync replication in which a stand-by copy of the database is required but need not be kept in sync up to the second. The stand-by copy lags behind the primary in terms of the data's currency, possibly only coming into sync when all activity against the primary copy is completed. From a DBMS perspective, the issues involved in deferred copy management include:

☐ Which mechanism is used

to maintain the stand-by copy?

☐ How current is the data in the stand-by? Is this option tunable?

☐ How is the stand-by brought in sync with the primary copy upon loss of the primary? Are some transactions inevitably lost if the primary system or network crashes?

☐ Can the stand-by reside remotely as well as locally?

☐ If the stand-by is constructed by downloads of the primary copy, can you ship incremental updates from the previous download or are only full downloads supported?

☐ At what granularity (entire database, table, and so on) is deferred copy management supported?

Once again, deferred copy management is a specialized architectural need for some applications. The most critical aspects here include the copy's currency and what occurs upon failure of the primary. In many cases, the application is willing to live with a slightly out-of-date version of the database.

OPERATIONAL ISSUES

Operational issues cover items especially critical to those who support or administer a DBMS product on a daily basis. Some of these issues refer to specific tools that are either necessary or would be helpful in systems administration. Others are more global issues that refer to the product's resilience and reliability and its effect on other system components.

OPERATIONAL INTERFACE

Critical to a systems administrator is the ability to know what's happening inside the DBMS. Typically, some sort of operational interface is provided as a "window" into the system. If no such tool is provided, troubleshooting and system tuning can become much more difficult. Some items to examine include:

☐ Is there a tool (preferably a graphical interface) to view the running system?

☐ Can reports be generated that outline the key system parameters and possible incorrect values?

☐ Are the operational tools aimed (optionally executed) at different experience levels (for example, novice versus expert)?

Performance is a key issue when choosing a DBMS product

☐ Is the concept of mapping functions to roles (for example, systems administrator versus DBA) in place?

☐ Do tools exist for start-up and shut-down of a DBMS or transaction-based system? Is the concept of a partitioned start-up and shutdown in place?

☐ Which tools are provided for life-cycle administration (installation, reinstallation, upgrades, and so on)? Are the operational tools included with the base product, purchased separately, or obtained through a third party?

Obviously, a systems administrator must regularly interact with the system's DBMS. This interaction should occur naturally. Does the administrator need to become an expert on the DBMS? What type of learning curve is required for new support staff personnel? Can administrators for one DBMS easily port their skills to another DBMS? Many of these questions can be answered by the functionality of the operational tools.

SYNCHRONIZATION

In many environments, it is desirable (if not mandatory) to have compatible versions of a DBMS product across all operating environments. This synchronization is necessary in a distributed environment, in which products must communicate with one another. It may also be necessary from a support standpoint since it is desirable for one person to be able to support a DBMS product in all environments. This concept is referred to as cross-environment synchronization. Some issues to analyze include:

☐ Where do your processors and operating systems fall in the vendor's porting cycle? (If a large gap exists between ports for two of your environments, you may find yourself with an upgrade in only one environment for some length of time.) How well does the vendor administer the product's portability across platforms?

☐ Does the product allow phased implementation and cutovers (for example, upgrading one processor in a distributed environment and another at a later time, rather than all at once)?

☐ Will tools from the prior release as well as third-party tools work with the DBMS engine's new release? (Again, this issue is critical in a phased cutover or a distributed environment.)

☐ Are multiple versions of the DBMS product supported under the same system image? (This feature may be useful during upgrades and beta tests.)

Ideally, the vendor provides a great degree of flexibility in upgrade procedures. In many cases, a flash cutover in a distributed environment is simply not feasible. The releases should be upwardly compatible and complement one another. In some cases, however, you will find that elaborate database or applications conversions are required to move from release to release. This fact makes matters quite complex, particularly in a distributed environment.

PERFORMANCE MONITOR

Performance is often one of the most important aspects when selecting a DBMS product. Good performance is comprised of many components, ranging from good planning and design to efficient processing. You can think of a performance monitor as the DBMS's life support system, which monitors and verifies that the system is achieving and maintaining desired performance levels. If it is not, the performance monitor points out problem areas and suggests or implements improvements. Some subjects to address in a performance monitor include:

☐ Does the product provide performance trending reports (response and transaction times, TPS rates, and so on)? Are they presented textually or graphically?

☐ Does the product indicate levels of CPU usage, DASD usage and accesses, and system idle time?

☐ Can the performance monitor isolate specific users and detect deadlock conditions? Can it detect who is waiting for record locks and who is holding them?

Can it determine runaway processes and terminate them?

☐ Does a trace facility exist that can report the number of times a transaction is processed, parsed, and executed?

☐ Does it provide heuristic guidelines for tuning and debugging aids?

☐ Does the performance monitor provide snapshot information and real-time information?

☐ Is it executed online or in batch?

☐ Which environment does the product run in (for example, host-resident versus PC-based)? Is it external or integrated into the engine?

☐ Is the performance monitor provided with the standard DBMS engine or is it sold separately? Is it or alternates provided by a third party?

You will probably find wide variations in the performance monitors from different vendors. Some will claim that several rudimentary commands make up the monitor, others will provide elaborate products, and other popular engines will offer third-party alternatives. Regardless, it is likely that a performance monitor of some kind will be a required component of any performance-oriented system.

CAPACITY PLANNING TOOL

Often, it is useful to test system workloads and performance characteristics prior to actual development. To do so usually requires some sort of capacity planning or applications-modeling tool. This tool will let a designer or administrator build a conceptual model of the applications system and provide its operational and performance characteristics and predictions as outputs. This type of tool can greatly reduce applications-development and design time. Some issues to examine include:

☐ Is such a capacity planning and modeling tool available with the product or from a third-party vendor?

☐ Can it predict the performance of utilities (backup, recovery, and so on) as well as transaction performance?

Unfortunately, this type of tool is not usually provided by DBMS vendors. Some third-party

Backup and recovery are crucial aspects of any system

vendors specialize in this field; however, they typically provide such tools for only the most popular DBMS engines.

RELIABILITY

This issue attempts to get at the product's production strength by relying on published statistics that the vendor should be able to provide, as well as specific features regarding fault tolerance and product reliability. Here, you must examine the vendor's ability to react and service problems in a crisis situation. Pertinent issues include:

☐ What is the rate of outages this product requires? Is a period of downtime required each day or can the product support an around-the-clock operation?

☐ What are the demonstrated mean time to failure and mean time to repair statistics from the vendor's customer base?

☐ Does the product provide any fault-tolerant capabilities?

☐ What is the vendor's commitment (on-site personnel, time commitments, and so on) to repair the product in a crisis situation?

In general, you must be confident that you are using a reliable product with a competent and committed vendor behind it. A vendor with a good track record has probably performed in mission-critical applications before. A vendor who cannot provide a track record may be just entering the market for such systems.

RECOVERY TOOLS

Database backup and recovery can be broadly defined as the ability to offload and reload the database to and from a reliable medium. This activity is an absolutely crucial aspect of any system, and provides an application with a necessary level of security against hardware errors and data corruption. You should examine backup and recovery from the standpoint of functionality, reliability, and performance. Specific

issues to consider include:

☐ Does the product provide utilities for backup and recovery or does it rely solely on operating system facilities?

☐ How and when is a full backup normally performed? To which type of archive medium can it be dumped (tape, disk, combination, and so on)?

☐ When backing up a database, how do you switch from one device to another when necessary?

☐ Are online backups (backups while the transaction system is active) supported and, if so, what is the implementation and its limitations?

☐ Is database mirroring supported? How does it come into play during backup and recovery cycles?

☐ Is there a fast dump utility (data import and export) you can use in lieu of the normal backup and recovery facilities?

☐ Are the backup and recovery tools tailored to address the skill levels of different users (novice, experienced, and so on)?

☐ Can a partial backup be performed or is a full backup always required?

☐ Does the product support incremental backups?

☐ Can multiple databases be backed up on the same machine in parallel?

☐ What performance measurements have been demonstrated by the product for backup and recovery (for example, time per MB, cost per MB)?

Backup and recovery are often overlooked as a significant issues in the selection process, and are often considered a necessary evil. The functionality and performance of these utilities (if they exist) can have a great effect on the ultimate operations of the application and, therefore, should not be overlooked.

JOURNALING

Journaling is the ability to record database updates to a medium so they can be used for database recovery in case data is lost or corrupted. Journaling is tightly integrated with the backup and recovery features of a product, as well as resilience to operating system and processor failures. In a typical scenario, when a large

amount of data is lost, you may recover from the last full backup and then apply the journal entries against the backup to bring the database up-to-date. When system failure occurs, the DBMS must check that all transactions in progress at the time of the failure have either been written to disk (if completed) or rolled back. Some key areas to explore include:

☐ Is transaction journaling required for a database or DBMS system? Can it be turned on and off?

☐ Can journaling be turned on and off at the table level (for example, journaling for select tables only)?

☐ Can tables be recovered from journals individually or must the entire database be recovered as a unit?

☐ How is journal management handled? Is a fixed number of journal files cycled and offloaded as necessary, or is a set of files expanded as needed?

☐ Is duplex journaling supported and, if so, how is it kept in sync? What is the journal file medium and what is it offloaded onto? (Obviously, the medium should be as reliable as possible.)

☐ Upon journal recovery, can all journal files be merged, applying only the latest version of a given record, rather than all interim updates to that record that may be in the journals?

☐ Is point-in-time recovery supported? Can you recover by user (all transactions except those executed by a given user) or by table (selective restore)?

☐ What is stored in the journals (for example, disk images or transactions to be reexecuted)?

☐ Are audit procedures in place to report on journal entries?

☐ How tunable is the journaling system (such as sizes and numbers of files, and so on)?

☐ What is the performance overhead associated with journaling? How has it been measured?

☐ Does the product have the ability to restore to a different physical medium or configuration (for example, in the case where a physical device must be changed)?

☐ After system failure, how is the database made consistent with the journals? Does the product support a warm or cold restart?

What medium is used to house the database and journal files?

Can you roll back the journals to the point of consistency and then backout and reissue the in-flight transactions? Can you roll back to a point in time? Are these features automatic, or can they be specified if necessary?

☐ What is the scenario for switching to stand-by copies of a database (local and remote)?

☐ How secure are the journal files and tapes?

Many of these issues may seem too technical to consider from an application standpoint; however, you must understand them to design and plan effectively for a successful system. Journaling and system recovery can be very complex issues but, since they can determine the degree to which data is consistent and up-to-date, they cannot be ignored.

MULTITIER RESTORATION

As distributed processing solutions become more and more popular, distributed administration takes on a high degree of importance. Multitier restoration addresses the ability for a networked environment to be treated as a single unit administratively. The following issues must be addressed in a distributed manner:

☐ Does the product permit system start-up and shutdown in a coordinated fashion among multiple sites?

☐ How is database backup, recovery, and system restart handled in a tightly integrated, distributed architecture?

Many DBMS vendors will leave such complex issues as coordinated administration for distributed environments to the hardware and network vendors. More and more, however, DBMS vendors are providing the networking software. The vendor who includes the capabilities to coordinate and administer such a complex environment will have a major advantage over the competition.

RELIABLE DATA STORAGE

Obviously, you must feel comfortable that any data an application stores in a database is stored in a safe and reliable place. More importantly, data that appears to have been committed in a transaction must actually be committed to the database. This concept is called reliable data storage. The important questions to ask include:

☐ What medium is used to house the database and journal files? How reliable and resilient is the medium to system crash?

☐ Does the transaction management architecture guarantee that any committed transaction will ultimately be reflected in the database? (This issue may be especially important for systems that store databases using the UNIX file system as opposed to raw disk slices. In this case, data can be written to system buffers, which may not make it to disk in the case of system crash.)

In general, reliable data storage is simply a requirement for any transaction-based system that must guarantee accuracy to the last transaction executed. If such security is not provided, the product is not a viable candidate for the system.

A TRICKY BUSINESS

As you can see from the many issues discussed over the past three months, DBMS selection can be a tricky business. I hope this series has provided you with new perspectives on this decision that will help you make more knowledgeable choices in the future. I encourage you to embellish and customize this list of issues by adding any key aspects of database science I may have overlooked or crucial product or vendor capabilities particular to your application. As your DBMS evolves, remember that you must continue to update this list over time. ■■■

The author wishes to thank all those who contributed to the AT&T Environment Selection Process (ESP) DBMS "brainstorming sessions," including the members of the DBMS Product Engineering District at AT&T IMS, as well as several valued customers who assisted. Special thanks to Chuck Wood, whose vision and dedication to the ESP DBMS Selection tool inspired this work.

Steven Caniano is the product manager for workgroup processing services for AT&T Information Management Services in Piscataway, New Jersey.

A PREVIEW OF COMING ATTRACTIONS

April may shower, but *Database Programming & Design* flowers! Don't miss our April 1992 issue:

- Repositories and Enterprise Data Architectures
- Distribution Protocols in DB2 v. 2.3: An Analysis
- Joe Celko's Report from the Latest ANSI SQL Standards Meeting
- How to Handle Change Management
- Guerilla Tactics for the Stealth DA

Plus Much More!

AntWorksTM THE SMALL AND MIGHTY EDITOR FOR ORACLE[®]



Imagine.....

A world where you could access and manipulate your ORACLE data without developing SQL, SQL*Forms[®] or C code.

That describes AntWorks. Finally, big productivity for a small price.

- Query, edit and browse data
- Load data into/from external files
- Build and modify tables

WITHOUT WRITING ANY CODE

It's menu driven, fully integrated to SQL*Forms, and can run on most platforms. AntWorks – the mighty editor that works harder and smarter for you.

Call today for a **free demo disk.**

MIACO
corporation

(800)877-0381 x225

MIACO, the MIACO logo and AntWorks are trademarks of Miaco Corporation.
All other products are registered trademarks of their respective holders.
Copyright © 1991 Miaco Corporation.

CIRCLE 27 ON READER SERVICE CARD



See us at Booth #321
Moscone Convention Center
San Francisco, CA
March 24 - 26, 1992

Finally, A CASE Tool That Works On Two Important Levels.

Your Professional Level And Your Frustration Level.



SILVERRUN – The Macintosh Leader In CASE, Now Brings Relief To The World Of Windows And OS/2.

Why settle for lower expectations, and less-than-professional results? Now, SILVERRUN™ meets your personal and professional needs for CASE technology that's high on results and low on frustration. Coming straight from rave reviews in the Macintosh world, SILVERRUN is packed with an abundance of advanced and powerful features to ensure that your modeling expectations will now be met.

Using the workstation platforms you already have — Windows, OS/2, and Macintosh — you'll leverage the latest data-driven, object oriented technology to construct large, comprehensive, real-world data and process models to meet any relational requirement.

Developed by expert modelers, SILVERRUN is an affordable, professional CASE tool designed to help you get the job done faster and better, not to get in your way. And with its open architecture and broad range of capabilities, SILVERRUN can enhance and complement your existing CASE environment — or be the only solution you'll ever need.

To arrange a hands-on demonstration that's on your level, call the CASE Professionals at **(800) 537-4262** today.

SILVERRUN delivers the leading edge features you expect from a professional CASE tool:

- Reverse data engineering from data specifications to ER models
- Automatic generation of graphical relational models from ER models
- Automatic generation of foreign keys, indexes, and SQL schemas
- Imbedded expert system technology to validate models using business rules in English
- User-controlled customization of modeling notation, dictionary reporting and transfer functions to suit organizational standards
- Hypertext-like dictionary navigation
- Intuitive, consistent graphics interface across modules and sharing of models across all three platforms (Windows, OS/2, Macintosh)

ADP 32

CSA Computer Systems Advisers, Inc.
50 Tice Blvd., Woodcliff Lake, NJ 07675
(800) 537-4262 In Canada (418) 687-5764

The Right Tool For The Right Job.

All names, products and services mentioned are the trademarks or registered trademarks of their respective holders.

CIRCLE 16 ON READER SERVICE CARD

REPOSITORY REPORT

How to keep your repository straight with version control

YOU'VE JUST installed a brand-new application that has consumed your life for the last year. Users love it because it satisfies all their requirements and requires no enhancements to meet the demands of your users' volatile business environment. It has no bugs and integrates effortlessly into your corporation's application portfolio. You're ready to face the challenge of developing yet another flawless application for your adoring business users. Ah, life is good.

Then the phone rings, jolting you back from your daydreams of happy users and finished systems. Your user begs you to include just one more little enhancement in the release that is currently being tested and wonders if the subsequent release could possibly be installed a month sooner. Welcome back to the wonderful world of applications maintenance.

An application seems to take on a life of its own when the first release is installed into production. Users are quite adept at identifying additional data and functions that need to be supported. New technology and design concepts become available so fast that they've often become obsolete before you've even had a chance to integrate them into your systems.

The need to develop evolutionary applications that are architected for flexibility and change is obvious. As Amdahl Corp. states when marketing Huron, its applications-development product: "An application is always complete, but never finished."

Techniques must be established to support the maintenance phase of the applications life cycle.

BY TERRY MORIARTY

Managing Application Evolution

Supporting change and version control may not be as fun and exciting as enterprise modeling and application design, but they are just as essential for successful information resource management.

Maintenance is the process of implementing changes and enhancements throughout the life of an application. Version control is a set of procedures and techniques that let multiple incarnations of the same application component exist simultaneously as the application evolves over time.

Version control has two levels: application component and application release. An application component is an identifiable constituent of an application. Examples include *entity*, *business function*, *program*, *database*, *data element*, and *test case*. In addition, an application component is documented as a repository object.

An application release, on the other hand, is an identifiable package of related application components that has been or is intended to be implemented. Multiple releases of the same application may be in development and production at the same time.

Application version control starts at the application component level. When an application component is created, its corresponding

repository object is assigned a version or revision number of zero. Whenever a change is made to the repository object, the version number increases by one. The unique name of the repository object consists of the name assigned by the repository user plus a specific version number. When the repository object is referred to by name only, the version with the highest version number is automatically accessed. However, a specific version can be identified by simply including its version number with the repository object name.

Consider the application component, *customer*, which describes an entity in the application's information model. Upon creation, it is assigned a version number of zero. As the definition of the application component evolves, new versions are created, as illustrated in Figure 1. The name "customer" identifies CUSTOMER_REV.2. However, the other two versions can be accessed by specifically identifying CUSTOMER_REV.0 or CUSTOMER_REV.1.

GUIDE refers to this type of version control as "serial." Each subsequent revision of a repository object creates a new object based

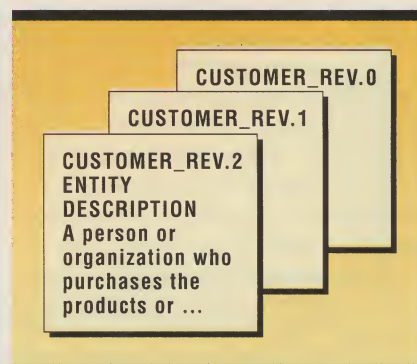


FIGURE 1. Repository object versions.

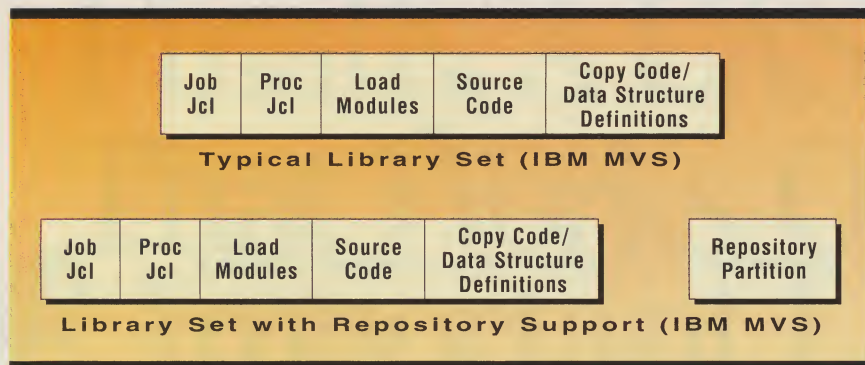


FIGURE 2. Incorporating the repository into the release library set.

upon the prior version. Each revision exists independently of the others, and the repository ensures that users access the correct version based on the version number included with the name.

THE SECOND LEVEL of version control is used to manage application releases. Release version control is nothing new to applications developers. Techniques have been implemented in most environments allowing new releases to be based upon prior releases, but evolve independently of them. The illustrations of this function in this column apply to the IBM mainframe environment, using Manager Software Product's DataManager repository. However, the concepts apply regardless of the technology used to implement the applications-development environment (ADE).

When the development of a new release commences, an environment is established through which the new versions of the application components are maintained. In the IBM MVS environment, a set of related libraries is created, as shown in Figure 2. Normally, application components are managed through libraries established for a specific purpose; that is, program source code and proc job control language (JCL) are maintained in separate libraries. As Figure 2 also illustrates, when the ADE is controlled through the repository, a repository partition must be created to house the repository object versions specific to the application release. With DataManager, the repository partition is implemented as a "status."

While the application compo-

nent resides in the appropriate library, the corresponding documentation is maintained in the corresponding repository partition. When the repository is used actively in production, the documentation in the repository is used to generate the actual application component, as illustrated in Figure 3. For example, a CASE tool uses the data structure Group Y, which is documented in the release's repository partition, to generate Y's COBOL data definition. The COBOL statements are stored as a member in the release's Copylib.

Likewise, the ADE's code generator should use the logic documented in Module X's minispec, which is maintained in the repository to generate X's COBOL source code. This source code is stored in the release's Sourcelib. Additional CASE tools should be available to generate database data-definition statements, proc and JCL, and other application components. The

libraries in a library set evolve through the development life cycle as a unit.

So far, we've only defined the library set of one application release. Let's expand the concept to handle multiple releases. Most applications are developed to support serial releases. The changes of the prior releases serve as the basis of a given release. Each release is intended to supersede the prior one. When application releases are serial, their library sets can be arranged into a library stack, as shown in Figure 4. The base library set contains all the application components for the entire application. Initially, the base is the library set for the first release of the application, but as the application evolves, it corresponds to the library set for the application components currently installed in production.

The library sets are placed in the stack in release order. A release library contains only those application components that have changed from the prior version. A repository must maintain repository object versions across release partitions by identifying which versions are referenced by a specific release. For example, when a change is made in the second quarter release of Module X, a new version (X_1) is created that is based on the production version.

Module X is modified again in the third-quarter release. Therefore, version X_2 is based on ver-

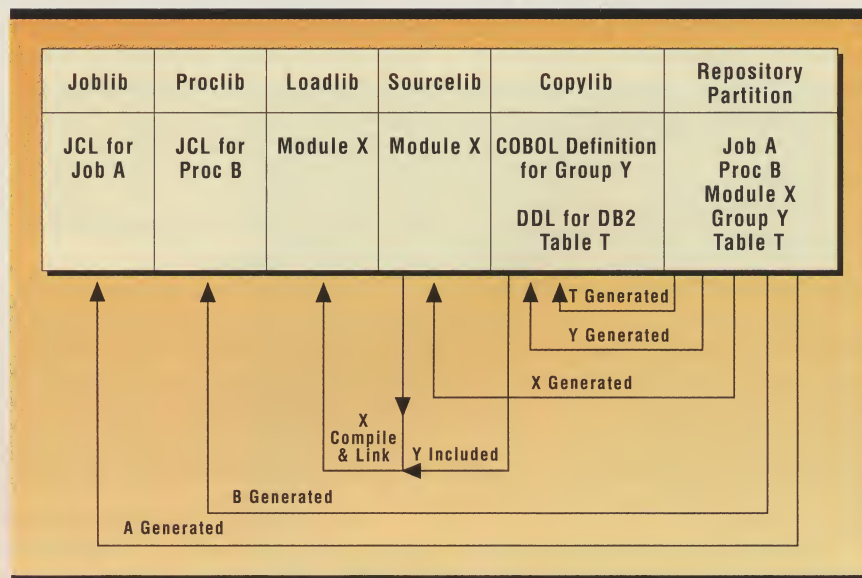


FIGURE 3. Interaction between repository and libraries.



FIGURE 4. Application release library sets.

sion X_1 from the second-quarter release. The library sets are concatenated, which allow a release to "see-thru" the library set stack and find the correct version of a repository object and its corresponding application component. For example, the third-quarter release references X_2 and Y_1, plus Z and B_1 from the first-quarter release and the production versions of A and T. DataManager's Advanced Status Facility provides support for the concatenation of repository partitions.

In an active ADE, application components are always generated from a repository partition into another library in the same release library set. Code is never generated from the third-quarter release repository partition into the first-quarter release Sourcelib.

WHEN A RELEASE IS installed into production, its library set is merged with the current production libraries. At this time, the version of the application components and repository objects maintained in the release's library set replace the older versions residing in the production-equivalent libraries. Likewise, new application components introduced into the new release are added to the base set. For fall-back purposes, it's prudent to retain a copy of the production-equivalent library set as the archive version before merging it with the current release.

Parallel version control allows several versions of a single application component to exist simultaneously. However, these versions

aren't meant to supersede one another. In the ideal world (in which the application is developed from an enterprise model), this type of version control probably isn't necessary. However, few corporations maintain this ideal application portfolio. Rather, legacy applications must also be managed and quite often manifest the following complications:

□ The same name is used to refer to entirely different application components;

□ The same application component uses the same name, but has been implemented differently in separate applications. For example, ACCT-STATUS-CD may use numeric codes in one application, while another uses alphanumeric codes;

□ Applications implement the same corporate definition for a repository object, but must stage the implementation of enhancements to that object independently;

□ The same application has

been customized to meet the needs of different customers;

□ Enhancements are being made to the corporate definitions, which are not yet ready to be implemented.

These situations require parallel version control in which independent library stacks can be maintained, as shown in Figure 5. Interrelated applications that integrate their release plans can share the same stack. A release stack can see-thru its own stack, but can't see-thru to other application's stacks.

Version control is an important management tool that provides the flexibility necessary to manage the application portfolio's evolution. A repository must provide support for versioning at both the repository object and release levels. Without version control, a repository's ability to control the ADE is severely limited.

Next month, we'll explore the role the repository plays in change management.

REFERENCES

- ANSI X3.138-1988, "Information Resource Dictionary Systems (IRDS)," sections 1-26 to 1-31, 1988.
- Murphy, J. "Beyond Information Modeling, Integrating a Repository into the Application Life-Cycle at Bank of America," 10th International Conference on the Entity Relationship Approach Presentation, October, 1991.
- GUIDE, "Repository Versioning Strategy Paper (GRP-222)," June, 1988.

Terry Moriarty is the manager of data administration for the Bank of California, a San Francisco-based financial institution.

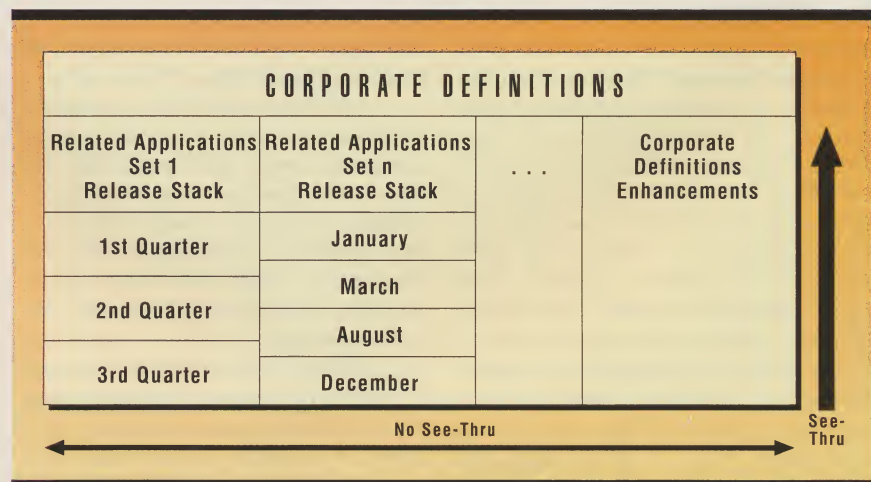


FIGURE 5. Application portfolio repository release stack.

FOR MANAGERS ONLY

Using different approaches to analyze historical data under hypothetical situations

ONE OF THE MORE intriguing aspects of data warehouse and DSS processing is the issue of rewriting history at the detailed, atomic record level once it has been correctly entered into the warehouse. (No, I'm not suggesting fraud, and yes, I'm serious.) Once the data warehouse has been built and the detailed, atomic data has been carefully transformed and loaded from the operational environment, DSS analysts—especially from the marketing and sales departments—have a strange habit of wanting to go back in time and change the way things actually happened.

Typically, organizations wanting to perform this type of analysis are the marketing and sales departments. DSS analysts in these organizations always ask such questions as, "If we had aligned the sales territories differently last year, how would sales have stacked up?" Or, "We want to consolidate sales territories, but what would have been the impact of this consolidation in the past?"

This type of "what if" analysis has two requirements: First, the past years' (or more properly, historical data in general) detailed sales data must have already been captured; and, second, it must be possible to use this detailed atomic data for determining how the territorial alliance would have affected sales. In addition, the detailed sales data should have one or more "discriminators" (such as ZIP codes) to be able to identify how sales territories would have been aligned had past circumstances been different. Given the appropriate historical and detailed data, you can answer these marketing and sales what if questions. But first, it's necessary to go back in time and "rewrite history."

BY WILLIAM H. INMON

Should We Rewrite History?

The most straightforward approach to this problem is to decide what the territorial changes should be and go back and adjust each of the detailed sales records with this new sales territory. However, while this approach is straightforward, it isn't without its various problems. Some of the drawbacks to this approach include:

□ Depending on how you make the changes to the detailed data, some of the data may be destroyed forever. In our example, the original sales territories may be destroyed. Say the algorithm for changing older territories is: "If territory=ABC and the corresponding ZIP codes are between 12345 and 23456, then reset territory to BCD." Depending on the alignment of territories and the discriminators used, it may be impossible to return the detailed data to its original state.

□ This method is costly in resources; not only for the initial modification of the territories, but for each successive iteration of what if analysis as well. If DSS analysts will be changing their minds frequently, then the resources inherent to this approach can be intimidating.

□ For a variety of reasons (including those already mentioned), this approach doesn't lend itself to iterative processing.

Another approach is to identify the discriminators in the de-

tailed data warehouse atomic record that are independent of the sales territory code. Typically, ZIP codes are used for this purpose. Then, create an index on discriminator(s) in the data warehouse. At this point, you may be thinking, wait a minute, this index is most likely going to be huge and will therefore require a lot of resources to build. This thinking is quite correct.

An alternative strategy to simply creating an index on the discriminator is to write a program that summarizes detailed atomic sales records by discriminator and some unit of time (such as daily, weekly, monthly, and so forth) prior to the creation of the index. This process is called "precondensation" or "preconditioning the data." For example, you could summarize sales data by ZIP code and month.

This summarization process requires more work than simply creating an index (unless you happen to have your sales data already sorted by discriminator and date). Ultimately, however, the precondensed data ends up being more compact and much easier to index. When you have the sales data summarized by discriminator and unit of time, you can index relatively efficiently.

In any case, whether you index the sales data at the detailed level on randomly organized data, or precondense the data and then index it, you end up with detailed sales data indexed by discriminator.

Without question, indexing and precondensing data requires a fair amount of resources. However, two points should be made here. Rewriting detailed history in any form is a resource-intensive process. Moreover, whether the index is built first, or the data is precondensed and then the index is built,

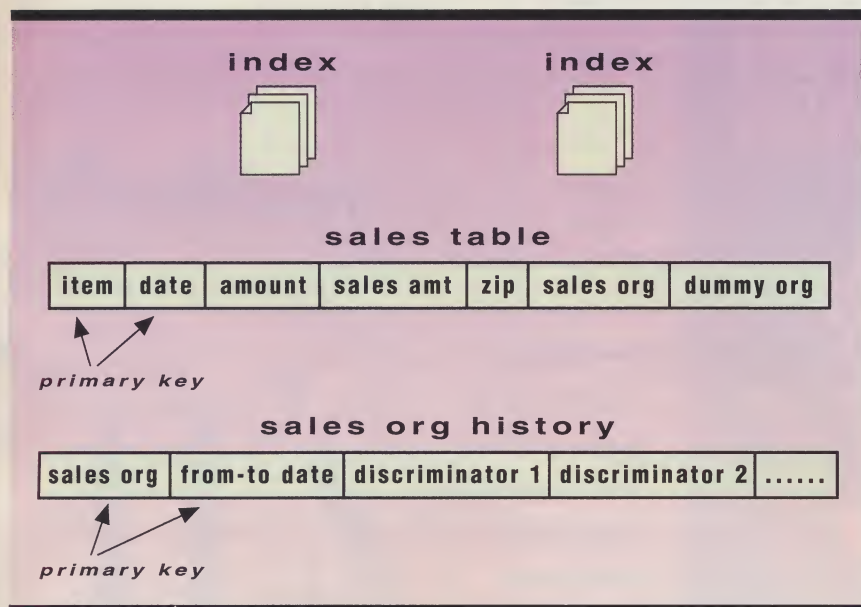


FIGURE 1. Data structures for rewriting history.

the data arrives at a state in which iterative analytical processing can be done as efficiently as possible.

Another way to precondition sales data is to add a field called DUMMYORG to each sales record. As you create this DUMMYORG field, you can load it with the initial territorial sales organization. In later processing, you can load the appropriate records with their "new" sales territory. Finally, you can create a new table called *organization history*. The purpose of this table is to trace the changes of sales territory by time. Given any moment in time, the DSS analyst can determine what the sales organization structure looked like by examining the contents of the organization history table. The table has three essential elements: the organization identifier, from-to dates, and the discriminators belonging to the sales organization during the time span identified by the from-to dates. Figure 1 shows an example of a sales and organizational history table.

Upon creating the extra tables and fields, marketing DSS analysts can have their cake and eat it too. The marketing department can analyze the sales as they really were or rearrange the sales districts by altering the contents of DUMMYORG to reflect the proposed changes. By using the organization history table, marketing DSS analysts can also return to any mo-

ment in time and reconstruct the marketing territories as of any moment in time. Such capabilities are the ultimate in rewriting history.

In many ways, the chosen discriminator—usually ZIP code—serves as a "lowest common denominator." Using the discriminator and other available data, DSS analysts can slice and dice data to their hearts' content.

Of course, the approach suggested here has both pros and cons. If the data is summarized by discriminator and a certain date, no further analysis below the level of summarization can be completed. For example, if the data is summarized by month, weekly analysis is impossible. Or, if data is summarized by ZIP code, no possibility exists to analyze data by street.

Summing data has its drawbacks, as is the case with any shift in granularity. On the other hand, summing sales data above the detailed level has major advantages. By summarizing sales data, the volume of data is often shrunk significantly. Therefore, the abilities to index data freely, access and analyze the data, and quickly create different iterations of analysis are enhanced by the reduction in data volume. As a result, designers must balance the expected use of the data with the resources consumed by it.

Other considerations exist as well. For example, adding the

DUMMYORG field may be a waste of space. If the DSS analyst wants to create a new organization only once, then the DUMMYORG field serves no real purpose. Only when the DUMMYORG field is loaded once and analyzed many times does it make sense to create it.

This column has addressed the issue of manipulating historical data to reflect a different organization and structure. If you think the trade-off is worth the cost of resources, then you can create and analyze "artificial historical data" fairly easily. However, you should keep the following rules of the road in mind:

- The original detailed data shouldn't be destroyed.

- Add dummy fields to accommodate an artificial structure.

- Restructuring the data should be considered when the artificial organization is created once and analyzed frequently and iteratively thereafter.

- Don't disturb basic discriminators once you identify them. (This rule is mandatory.)

- Summarize data up to the lowest common denominator prior to indexing in order to save considerable resources.

- An index is necessary to perform reorganization iteratively.

- If only a one-time analysis is desired, then a simple join of data into a working set may be a good answer.

- A historical record of organizational unit by discriminator is useful.

A final related question is, when should original detailed data in the warehouse be altered? For example, three years after the fact, is it ever appropriate to change the value of a sale, item sold, or location? As a rule, unless the detailed sales data has been incorrectly recorded, it doesn't make sense to alter data. However, a few legitimate exceptions may exist. Consider what happens when the post office changes existing ZIP codes. When this case occurs, it may make sense to go back and alter old ZIP codes to make them conform to newer classifications. ■■■

William H. Inmon, a database author and lecturer, works at Forest Rim Technology in Littleton, Colorado.

CALL 1 (800) 2DB-EXPO
FOR YOUR FREE EXPO PASS OR
MORE CONFERENCE INFORMATION!

REAL SOLUTIONS NOW



JOIN OVER 15,000
DP PROFESSIONALS AS
THEY SEE AND TEST THE
LATEST IN DATABASE AND
DEVELOPMENT TOOLS

DB/EXPO '92 tackles today's most critical issues
and problems. Five concurrent programs explore
pivotal topics, including:

- DATABASES
- INTEROPERABILITY
- OPEN SYSTEMS
- DOWNSIZING/RIGHTSIZING
- CLIENT/SERVER
- CASE TOOLS
- OBJECT DATABASES
- DISTRIBUTED AND RELATIONAL DBMSs
- MULTI-MEDIA
- NETWORKING and more

The Exposition floor includes over 400 exhibits,
featuring the industry's hottest products. See what's
new with trend-setters and industry
giants like Apple, Borland, Computer
Associates, DEC, Fox Software, H-P,
IBI, IBM, Microsoft, Novell,
Oracle, Teradata
and Wordtech.

THE WORLD'S LARGEST DATABASE AND TOOLS EXPOSITION AND CONFERENCE

DB/EXPO '92 -- THE DP PROFESSIONALS CHOICE!

MARCH 23-27, 1992 • MOSCONE CENTER • SAN FRANCISCO, CA
1-800-2DB-EXPO • 24 HOUR REGISTRATION VIA FAX: 1 (415) 941-2066

YES, I WANT TO ATTEND DB/EXPO '92!

Please send me: ☐ FREE EXHIBIT PASSES
☐ Information on conference sessions
and early registration DISCOUNTS

Name _____
Title _____
Company _____
Address _____
City, State and Zip _____
Telephone _____

FAX _____



Complete and return this coupon to:
NDN ENTERPRISES, 289 S. San Antonio Rd,
Suite 204, Los Altos, California 94022
1 (415) 941-8440 • 1 (800) 2DB-EXPO

24 HOUR REGISTRATION VIA FAX: 1 (415) 941-2066

ADVERTISER INDEX

	PAGE NO.	CIRCLE NO.
Bachman Information Systems	Cov3.....	20
Benchmark Technical Services Inc.	12.....	7
Bez Systems.....	74.....	17
BMC Software Inc.	13.....	8
Borland Int'l Inc.....	7.....	4
Brownstone Solutions.....	1.....	2
Candle Corp.....	4.....	3
CDB Software.....	27.....	15
Chen & Associates Inc.	23.....	13
Computer Associates Int'l.....	Cov4.....	21
Computer Systems Advisors Inc.	66.....	16
Compuware Corp.....	20.....	12
Concentric Data Systems Inc.	38.....	23
Datura	8.....	5
Database World.....	76.....	19
Digital Consulting Inc.....	54.....	26
Evergreen CASE Tools.....	52.....	25
Gupta Technologies Inc.....	Cov2.....	1
Information Builders Inc.	35.....	22
Infotel Corp.	19.....	11
International DB2 Users Group.....	53.....	*
Logic Works	17.....	10
Miaco Corp.....	65.....	27
Microsoft Corp.....	28-29.....	*
Morgan Kaufman Publishers	75.....	18
Norm DeNardi Enterprises.....	72.....	*
Oracle Corp.....	2.....	*
Platinum Technology Inc.....	40-41.....	24
Reltech Products Inc.	10.....	6
SAS Institute Inc.	9.....	*
Specialized Software Int'l.....	14.....	*
SQL Solutions Inc.	24.....	14
Uniface Corp.....	33.....	9

The index on this page is provided as a service to our readers.
The publisher does not assume any liability for errors or
omissions.

Need more
information?

IT'S
FREE

1

Fill in name and address information
requested and answer the questions.

2

Circle the numbers for
products you are interested in.

3

Mail today,
or FAX to 413/637-4343.
Literature will be mailed to you
free of charge.

DATABASE
Programming & Design

FREE PRODUCT INFORMATION

Please Print

☐ Ms. ☐ Mr.

DATABASE READER SERVICE CARD

Name

Title

Company

Address

Dept. / Mail Stop

Phone

City

State

Zip

50KR4

A. Requested products planned for purchase within:

1. ☐ 0-6 months 2. ☐ 6-12 months 3. ☐ 12-18 months 4. ☐ Reference only

B. Number of employees at your work location:

5. ☐ under 500 6. ☐ 500-1000 7. ☐ over 1000

C. PLEASE HAVE A SALES PERSON CONTACT ME REGARDING THESE PRODUCT NUMBERS:

(1-50)

(51-100)

(101-150)

(151-200)

1	2	3	4	5	51	52	53	54	55	101	102	103	104	105	151	152	153	154	155
6	7	8	9	10	56	57	58	59	60	106	107	108	109	110	156	157	158	159	160
11	12	13	14	15	61	62	63	64	65	111	112	113	114	115	161	162	163	164	165
16	17	18	19	20	66	67	68	69	70	116	117	118	119	120	166	167	168	169	170
21	22	23	24	25	71	72	73	74	75	121	122	123	124	125	171	172	173	174	175
26	27	28	29	30	76	77	78	79	80	126	127	128	129	130	176	177	178	179	180
31	32	33	34	35	81	82	83	84	85	131	132	133	134	135	181	182	183	184	185
36	37	38	39	40	86	87	88	89	90	136	137	138	139	140	186	187	188	189	190
41	42	43	44	45	91	92	93	94	95	141	142	143	144	145	191	192	193	194	195
46	47	48	49	50	96	97	98	99	100	146	147	148	149	150	196	197	198	199	200

☐ Please send me one year (12 issues) of DATABASE PROGRAMMING & DESIGN. Please bill me; U.S. \$37. Canada/Mexico \$43; All other foreign \$52 surface, \$77 air mail.

Comments:

March 1992 issue - expires June 30, 1992

FREE PRODUCT INFORMATION

Please Print

☐ Ms. ☐ Mr.

DATABASE READER SERVICE CARD

Name

Title

Company

Address

Dept. / Mail Stop

Phone

City

State

Zip

50KR4

A. Requested products planned for purchase within:

1. ☐ 0-6 months 2. ☐ 6-12 months 3. ☐ 12-18 months 4. ☐ Reference only

B. Number of employees at your work location:

5. ☐ under 500 6. ☐ 500-1000 7. ☐ over 1000

C. PLEASE HAVE A SALES PERSON CONTACT ME REGARDING THESE PRODUCT NUMBERS:

(1-50)

(51-100)

(101-150)

(151-200)

1	2	3	4	5	51	52	53	54	55	101	102	103	104	105	151	152	153	154	155
6	7	8	9	10	56	57	58	59	60	106	107	108	109	110	156	157	158	159	160
11	12	13	14	15	61	62	63	64	65	111	112	113	114	115	161	162	163	164	165
16	17	18	19	20	66	67	68	69	70	116	117	118	119	120	166	167	168	169	170
21	22	23	24	25	71	72	73	74	75	121	122	123	124	125	171	172	173	174	175
26	27	28	29	30	76	77	78	79	80	126	127	128	129	130	176	177	178	179	180
31	32	33	34	35	81	82	83	84	85	131	132	133	134	135	181	182	183	184	185
36	37	38	39	40	86	87	88	89	90	136	137	138	139	140	186	187	188	189	190
41	42	43	44	45	91	92	93	94	95	141	142	143	144	145	191	192	193	194	195
46	47	48	49	50	96	97	98	99	100	146	147	148	149	150	196	197	198	199	200

☐ Please send me one year (12 issues) of DATABASE PROGRAMMING & DESIGN. Please bill me; U.S. \$37. Canada/Mexico \$43; All other foreign \$52 surface, \$77 air mail.

Comments:

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO.886 PITTSFIELD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DATABASE
Programming & Design

Reader Service Department
P.O. Box 5063
Pittsfield, MA 01203-9634

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO.886 PITTSFIELD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DATABASE
Programming & Design

Reader Service Department
P.O. Box 5063
Pittsfield, MA 01203-9634

Need more
information?

**IT'S
FREE**

See reverse side for an
easy way to get more
information on the products
and services advertised in
this issue of **DATABASE
PROGRAMMING & DESIGN**

DATABASE
Programming & Design

BIRDS OF A FEATHER

Organizing a productive users group entails a lot more than just getting together

CASE IS A HANDY acronym that means Computer Assisted (or aided) System (or Software) Engineering. However, not only is CASE misunderstood by many people, but its implementation also presents various difficulties. In fact, CASE implementation requires surmounting a host of barriers as well as overcoming corporate resistance. This dilemma is the major reason why CASE success stories are so few and far between: Just because a company manages to implement one CASE product successfully, it's not assured that the next project will be as successful, or even that CASE will be used again. Yet, in the few instances when CASE methods have been used successfully and metrics existed to measure the productivity differences in using them, the results have often been dramatic. Therefore, people have come back again and again to take a stab at CASE.

To help those braving the twisted path that leads to successful CASE project implementation, a group of us formed the San Francisco Bay Area CASE Users Group (SFBACUG) in October 1989 in order to share information and experiences. The first meeting, publicized by the good auspices of Arthur Young (presently Ernst and Young, one of the biggest boosters of CASE users groups nationwide), was held at the California State Automobile Association building in San Francisco. Approximately 150 people attended, and a core group was formed to create a board of directors. Probably because of a lack of good sense, I allowed myself to be named president; a post I've held for two years now. And

BY DAVID PLOTKIN

A CASE for Users Groups

the group has grown and learned a lot during these past two, informative years.

A lot of what we've learned has to do with running a technically oriented users group. The first members of the board were volunteers but weren't quite sure what they were volunteering for. Since starting a new users group is a tremendous amount of work, not everyone who volunteered was up to the task, and a few changes took place in the ranks of the board during the first year. Therefore, it's important to define each office or position's tasks up front. An estimate of the time demands of each job is also helpful because it makes members understand exactly what they are committing to *before* they volunteer.

SFBACUG's board of directors consists of a president, vice president of membership, secretary, treasurer, and vice president of programs and events. The latter position is shared—it turns out that arranging six meetings a year takes quite a bit of time and effort. We've found that the best people for this job are consultants because they have lots of contacts in the industry, know how to go about borrowing room space, and are generally well organized. Unfortunately, consultants are often extremely

busy earning a living. Therefore, sharing the responsibility seems to work pretty well.

IT'S IMPORTANT NOT to forget the busy work. A tremendous number of tasks such as folding, stuffing, and mailing meeting flyers, duplicating and mailing newsletters (don't forget to appoint a newsletter editor), and so forth must be completed by someone in order for the group to function. It's possible to hire a commercial outfit to undertake this work, but we hired a responsible high school student (the treasurer's son) who was willing to do it. We pay him for his efforts, and we're very glad to have his help.

Of course, the matter of money must also be considered. Dues to cover expenses and hold meaningful programs and agendas are essential. The value offered to members should be worth what you're charging, and you need to come up with a dues structure that isn't too onerous. For example, most of the members of the SFBACUG work in corporations. Therefore, it's far easier for them to request a single check once a year for membership than to pay on a per-meeting basis. CASE users groups in other cities structure the collection process differently based on the group's make-up, but we've found that a single yearly payment also makes it easier to estimate a budget accurately.

Other sources of money are available, but you should be careful in these instances. For example, some of the larger tool vendors might be eager to make a presentation to the group and would pay

for the privilege. In San Francisco, however, we haven't chosen to approach vendors in this way, even though other CASE users groups have decided to take advantage of the opportunity.

Most importantly, we had to decide what we wanted the SFBA-CUG to accomplish. One thing it *isn't* is a replacement for tool vendor users groups, which are groups of tool users who get together to discuss the ins and outs of that tool. Therefore, we haven't and don't intend to hold sessions on the various tricky ways to use entity-relationship diagrams in IEW or IEF. Instead, we've concentrated on the general issues of using CASE. We've tried to provide programs that will ease the way for our members, letting them know what to expect, explaining what to do about hurdles, and offering a way to learn from each others' experiences.

Because many of our members are just starting out in CASE, some of our meetings have addressed the "how to's" of actually

One thing SFBACUG isn't is a replacement for tool vendor users groups

using it. We have presented step-by-step guidelines for projects, and members who have completed successful (or not-so-successful) projects have shared their experiences. We've also discussed what methodologies are, how they affect an organization, and how to formulate one that works.

Another series of topics has centered around how to convince an organization that CASE is a good investment. CASE is a new way of doing things, and many organizations are resistant to large changes. Managing organizational change becomes important in handling this resistance. Measuring the changes that result in using

CASE is also important. In some organizations, a gut-level feeling that CASE won't increase productivity enough to warrant a large investment in equipment and training pervades. The question then arises: How do you measure your productivity before and after you try CASE? What exactly do you measure?

Quality is another hot issue in the group. Most people believe that CASE will improve the quality of the developed system. However, a lot of factors contribute to final quality, not the least of which is the rigor with which you follow the life cycle to develop a system. Quality improvement becomes more than just a passing interest to those seriously contemplating using CASE to develop their next mission-critical system.

The group also discusses general topics, such as the future of CASE (Darlene Brown of the Gartner Group gives a great presentation on this subject), information on international CASE symposiums, and many other important issues.

DB2 Performance Management & Capacity Planning Workshop with Dr. Boris E. Zibitsker



the first hands-on workshop for MVS professionals, with or without prior DB2 experience, who want to effectively manage a DB2 environment.

Performance Analysts, Capacity Planners, System Programmers, and Technical Managers can take advantage of a powerful combination of THEORY and PRACTICE.

During this workshop participants will:

- * design and create DB2 databases
- * create and run SQL queries
- * monitor performance using DB2 Performance Monitors
- * model DB2 applications' performance using CRYSTAL™ Performance Evaluator, and BEST/1 - MVS
- * learn how to plan for distributed databases

This 5 day Workshop is held every month at our location in Deerfield, Illinois. Special arrangement can also be made for on-site workshops.

Mark your calendar NOW for one of our spring workshops:

March 2 - 6, 1992
April 6 - 10, 1992
May 4 - 8, 1992

For registration or more information call
BEZ Systems, Inc., at
(708) 940-1010.



CRYSTAL, Performance Evaluator and BEST/1 - MVS are registered trademarks of BGS Systems, Inc.

CIRCLE 17 ON READER SERVICE CARD

W_RITE F_OR U_S

We are looking for experienced professionals to write practical, how-to articles for *Database Programming & Design*.

Future issues will include topics such as:

- Improving relational and nonrelational DBMS performance
- Database design and data modeling
- Using PC DBMSs as front ends
- CASE and life-cycle development aids
- Applications development with productivity tools
- DEC and UNIX databases
- Distributed database, client/server, and co-operative processing architectures
- DB2, SQL/DS, IMS, IDMS, Sybase, Supra, Oracle, and other database products

If you have an idea, a draft, an article, or a request for writer's guidelines, we want to hear from you!

Theresa Rigney
Database Programming & Design
600 Harrison St.
San Francisco, CA 94107
(415) 905-2482

Because big names in the CASE industry come to San Francisco fairly frequently, we've managed to get a few of them to speak on occasion, including Vaughn Merlyn of Ernst & Young.

One thing that has really contributed to the success of our meetings is holding them in conjunction with CASE-related events. CASE World, DB/Expo, and AppliCASE are large shows in the San Francisco Bay Area that draw big crowds. The promoters of these shows have been agreeable about providing meeting rooms, speakers from their roster, and sometimes even meals after our meetings. They've also distributed our meeting announcements, which has saved us time and money. And, of course, our members received a discount for the show. In return, we've made our mailing list available to them to help promote the conference.

A network of CASE users groups spans across the nation and Canada, with groups located in most major cities. The International

CASE Users Group (ICUG) is an umbrella organization that is attempting to create a more cohesive atmosphere among the individual CASE users groups. This cohesion would help make sharing information easier. Furthermore, many of the newer groups have the same questions that older groups have already tackled. Groups could also share information on speakers and handle referrals to local users groups. In addition, chapter starter kits are now available to new chapters. To obtain more information on ICUG, you're welcome to call (213) 688-5466.

Unfortunately, ICUG's ambitious plan has been slowed by a lack of funds (corporations are now being solicited for contributions) and the fact that many local CASE users groups don't see the need for an umbrella organization and affiliation with ICUG. I suspect, however, that the huge discount offered to members of affiliated chapters attending the CASE Users Group Conference in Atlanta (October 12 to 14, 1992) will

convince most chapters that affiliation is a good idea. San Francisco was one of the first chapters to join ICUG, and we've already seen a benefit from sharing information with the groups in Los Angeles, Boston, Seattle, and Charlotte, North Carolina.

Being a member of a CASE users group is a rewarding experience. It helps you learn a lot, avoid a host of problems, meet new people, and hear about new job opportunities. In San Francisco, we have almost 100 members, and are starting a big membership push to encourage the approximately 400 people on our mailing list to join. If you use CASE and live in or near a big city, check out your local CASE users group. If you live near San Francisco, I'll see you at our next SFBACUG meeting. ■■■

David Plotkin is an information systems analyst working in data administration and modeling at Chevron Corp. in San Ramon, California. He is also the president of the San Francisco Bay Area CASE Users Group. For more information on SFBACUG, please call (510) 901-3528.

MORGAN KAUFMANN ANNOUNCES

A Guide to Developing Client/Server SQL Applications

Setrag Khoshafian (Portfolio Technologies, Inc.), Arvola Chan (Versant Object Technology), Anna Wong (Borland), and Harry Wong (Nomadic Systems)

March 1992; ISBN 1-55860-147-3; approx 600 pages; paper; \$32.95

"I highly recommend this book... it gives excellent treatment to a broad range of database issues, and its coverage of database server products reflects the deep knowledge of the authors. The information in this book will save professionals many years in research and trial-and-error learning."

Richard Finkelstein, Performance Computing

"If you are moving to a client/server environment, this book is the place to start. This book provides a clear,

readable description of the major client/servers in a single source. It ought to save you several months of research time at a minimum."

Joe Celko, Consultant and Columnist for DB Programming and Design

"This book is required reading for anyone interested in database client/server technology."

Robert Orfali, Author of Client/Server Programming with OS/2

This book is a practical introduction to DB management for applications programmers, DBA's and workstation users entering the client/server environment. It includes a tutorial on DB management concepts, explained with standard SQL, and offers a comprehensive survey of client/server technology with case studies comparing the capabilities and interfaces of dominant database servers such as EE, RDB, Oracle and SQL Server.

Includes:

- Features and performance of commercial DB servers
- Individual SQL dialects compared to the standard
- Server administration requirements
- Benchmarking techniques
- SQL applications in heterogeneous environments
- Application programming interfaces to the servers
- Future directions for SQL Client/Server Architectures

A Guide to Developing **Client/Server SQL Applications**

Setrag Khoshafian
Arvola Chan
Anna Wong
Harry K. T. Wong



FORTHCOMING!

A Companion to the SQL Standard

Jim Melton and Alan Simon

Fall 1992; ISBN 1-55860-245-3; approx 450 pages; paper; \$34.95

A thorough guide and tutorial for SQL 2, co-authored by Jim Melton, Editor of the new emerging ANSI/ISO standard for 1992.

TO ORDER IN THE US or Canada, call toll free 1-800-745-7323

Or, send check, money order or Amex, VISA or MC authorization (acct. no., name on card and expiration date) to Morgan Kaufmann, 2929 Campus Drive, Suite 260, Dept. CH, San Mateo, CA 94403. Include shipping & handling (US surface rate: \$3.50 1st volume, \$2.50 each additional; Int'l surface rate: \$6.50 1st volume, \$3.50 each add'l). CA residents add sales tax. Int'l customers add 10% to your order. Fax orders to (415) 578-0672. Phone: 415-578-9911. Email: morgan@unix.sri.com

CIRCLE 18 ON READER SERVICE CARD

DATABASE PROGRAMMING & DESIGN

Where Can You Find The Latest In Database and Client/Server Technologies?

More Education
For Your
Time & Money
Over 100 speakers!

The Best Database Event in the World Gets Better.



DATABASE WORLD

Conference & Exposition
June 29 - July 1 • Boston

4 DATABASE WORLD Conferences PLUS
2 CLIENT/SERVER WORLD's Conferences
= 6 Educational Opportunities In One Event!!

DATABASE WORLD

- Object-Oriented Conference
- Database Management Technologies Conference
- PC Database Conference
- CSP/DB2 Developers Conference

CLIENT/SERVER WORLD

- Client/Server DBMS Conference
- Client/Server Applications and Networking Conference
- Product Lab

And a huge exposition with the leading
hardware & software companies!

NEW!

CLIENT/SERVER WORLD!



Call 508-470-3880
For Detailed Conference
and Exhibitor Information

CIRCLE 19 ON READER SERVICE CARD

DATABASE LIBRARY

A reference to help OS/VS programmers migrate to COBOL II

TECHNICAL AUTHORS are possibly the most critical book reviewers. They demand that the writing be clear, correct, concise, and interesting. Well, I'm a technical author, and I've found all of these qualities in *MVS COBOL II Power Programmer's Desk Reference* by David Shelby Kirk. Notwithstanding the silly Game Boy-esque title, this book is easily the best COBOL II technical book I've ever read. It's much more than a desk reference, and Kirk explains COBOL II topics in an intelligently organized, conversational manner. Nearly all of COBOL II's critical elements are covered, including terms and concepts, benefits, OS/VS features unavailable in COBOL II, modified OS/VS COBOL operations, new COBOL II options, debugging and efficiency considerations, compile, job control language (JCL) requirements, CICS and DB2 issues, and more.

Kirk has obviously been working with COBOL II for quite a while and has probably taught classes in the subject, so he knows how to present concepts and technical material. He annotates all aspects of the language throughout the book and liberally doles out seasoned, professional opinions and practical hands-on advice for which you would typically have to pay consultants huge amounts of money.

Almost every aspect of COBOL II is covered first technically, then subjectively. The subjective discussions lean toward a multitude of important topics: practical usage, benefits, efficiency, programming clarity, trade offs, and so on. You'll rarely find this level of in-depth, qualitative coverage in the usual technical data processing books. For example, Kirk sug-

The Many Facets of COBOL II

gests placing all accumulators—or switches that affect a particular logic path—into a single 01-level entry and using the INITIALIZE verb. He then states, "I've been criticized by a few technicians for recommending INITIALIZE since it generates a few more machine instructions than the proper number of MOVE statements. I mention it here because these technicians may also work at your shop. If machine cycles become more expensive than your time, I'll change my views, but I think the industry passed that mark two decades ago."

The book is geared toward professional OS/VS programmers who are migrating their shops to COBOL II. The orientation of the chapters as well as their content make liberal use of comparisons between the traditional and modern approaches to things. This orientation could be considered both a strength and a weakness of the book, however. If your shop supports both COBOL and COBOL II programs, I can't think of a better way to present this information—programmers will undoubtedly be given OS/VS programs to maintain; probably within the next three to five years. On the other hand, if your shop has completely done away with OS/VS technology, this material is superfluous.

A possible downside to Kirk's approach concerns stylistic coding bias. Although he is careful to add disclaimers on many of the stylistic considerations throughout the

text, new COBOL II users will probably be influenced to code "à la David Kirk," which could lead to problems with COBOL II standards in some shops. However, you can't have your cake and eat it too. Readers pay authors for facts and opinions. And the opinions are probably more important than the facts in the long run. If you have strong opinions that have served you and others successfully, you might as well "stick to your guns." Kirk doesn't ride the political fence in this book. Rather, he expresses the good, the bad, and the ugly of COBOL II.

As a summary, Kirk presents COBOL II coding guidelines, which contain abbreviated versions of many of the previously mentioned concepts and techniques, as well as COBOL programs, related publications, and references for COBOL II, compile options, and JCL.

My only complaint with this book is its length—it's too short! Not counting the appendixes, the book is only 225 pages long. I could have benefited from more detailed examples in the DB2 and CICS sections, and an all-encompassing sample program or application would have tied the hundreds of COBOL II's detail aspects together. Instead, Kirk includes short yet clear figures and examples. Picky, picky, picky.

However, the strengths of this reference outweigh any weaknesses. If you're looking for a COBOL II book that qualitatively describes all facets of the language and helps you migrate your applications from OS/VS technology, then look no further.

MVS COBOL II Power Programmer's Desk Reference by David Shelby Kirk. QED Information Sciences Inc., 1991. ISBN 0-89435-305-5, 310 pages, \$34.95.

—Jonathan Sayles

PRODUCT WATCH

New maintenance, editing, and data conversion tools spring into action

GET THE BUGS OUT WITH ANTWORKS

Miaco Corp. has released AntWorks, a small yet powerful editor for Oracle databases. The product increases productivity by eliminating the need for users to write code for many commonly used procedures. According to the company, AntWorks is an editor, browse utility, data transfer facility, table manager, and an alternative to SQL* Loader in a single product. In addition, the product provides an intuitive, menu-driven interface for users to accomplish their database editing goals.

The AntWorks editor is portable and will run on most platforms in which Oracle resides. Pricing of AntWorks for a single DOS user starts at \$275, with multiuser environments priced by platform and number of users.

Miaco Corp., 6300 S. Syracuse Way, Ste. 415, Englewood, CO 80111, (303) 741-0381.

CIRCLE 101 ON READER SERVICE CARD

INTERSOLV RELEASES DESIGN RECOVERY

Intersolv has begun shipping version 1.0 of Design Recovery for DOS, a comprehensive, maintenance and redevelopment solution for PC-based developers. The product automates the recapture, reanalysis, and redesign of existing applications. With Design Recovery's access to information and its open desktop development environment, developers can analyze and document existing systems.

By automatically scanning information into a centralized LAN-based repository, Design Recovery automates the capture of existing

systems information into a CASE environment. From its repository, information for use in redevelopment can be accessed directly by Design Recovery's own analyzers, Intersolv's Excelerator analysis and design tool, or Intersolv's APS application generator. In addition, the product's open architecture provides for exchange of data with other CASE tools via IBM's External Source Format or the Electronic Industries Association-approved CASE Data Interchange Format.

Design Recovery reclaims design specification from code to produce high-level graphical representations of data, process, and user interface information, which facilitates further understanding of an application. The product is able to capture individual programs, groups of programs, or entire systems. Data is represented in data model diagrams, and screens and programs are represented in an "Invocation Diagram" format, providing detail on program control as well as the data flow throughout a program, even if the program is unstructured.

Design Recovery for DOS is priced at \$6,500 per user. Minimum requirements include an IBM PC or compatible running DOS 3.3 or higher, 2MB memory, and 5MB free disk space for swapping.

Intersolv, 3200 Tower Oaks Blvd., Rockville, MD 20852, (301) 230-3200.

CIRCLE 102 ON READER SERVICE CARD

THREE NEW RELEASES FROM BMC

BMC Software Inc. has announced the general availability of three new products: Trimar DEDB Secondary Index Facility, Opertune version 1.1 for DB2, and Copy Plus

for DB2 version 3, release 1.

Trimar DEBD Secondary Index Facility is BMC's latest addition to its Trimar Fast Path Series and provides secondary indexes for data entry databases (DEDBs), giving application programs an easy method for retrieving data from a DEDB in an order other than primary key sequence. Users must no longer spend valuable time developing their own programs for implementing DEBD secondary indexes. The product saves time and money, is easier to maintain, offers greater data integrity, and lets users' applications be up and running sooner than user-developed programs.

Opertune version 1.1 for DB2 allows for the dynamic modification of DB2 subsystems without interruption to users. These changes are categorized into parameter elements or operational assist functions. The product has the ability to terminate runaway threads without cancelling a user's address space, which means that the user's IMS and CICS regions, as well as TSO user IDs, remain intact. Other operational assist functions include active log manipulation and the freeing of table spaces for utility maintenance.

Opertune eliminates tuning restrictions through a technique that causes DB2 itself to make many of the requested changes upon demand. Significantly, these changes are accomplished without USER-MODs or dynamic hooks to DB2.

Copy Plus for DB2 version 3, release 1 is a high-speed copy utility for DB2. The new version offers increased performance and new functionality, including the ability to make up to four concurrent image copies. The product is designed and intended to supplant

IBM's DSN1COPY utility.

Additional enhancements to the product include support for 32,000 pages, multiple copy statements, and multidata set nonpartitioned table spaces. BMC has also reduced virtual storage requirements below the 16MB line. In addition, the product includes table space validity checking, which ensures that the copied pages have the correct internal formats.

Trimar DEDB Secondary Index Facility is priced by CPU level and begins at \$18,000 for a perpetual license. Prices for Opertune version 1.1 for DB2 start at \$14,500 for a perpetual license. Copy Plus is priced by CPU level and begins at \$12,000.

BMC Software Inc., P.O. Box 2002, Sugar Land, TX 77487, (800) 841-2031.

CIRCLE 103 ON READER SERVICE CARD

ENHANCED RDBMS/ 4GL CAPABILITIES

CompuServe Data Technologies has announced the immediate availability of Version 9 of System 1032, the company's RDBMS/4GL system for Digital Equipment Corp.'s VAX/VMS family of computers. The Version 9 release expands System 1032's data handling, user interface, and VMS integration capabilities. The COLLECT JOIN function enhances System 1032's ability to manipulate massive databases by extending its capabilities of joining multiple related datasets (tables). The command window provides a new window-oriented user interface to access data more easily and flexibly. Additionally, Version 9 builds on System 1032's VMS integration with enhanced VMS services, memory, and disk usage tools.

Expanding on existing database join facilities and the union collections already in System 1032, the new join collections provide additional options for combining datasets into single logical records. Users can collect datasets that weren't originally designed to be part of a database view without performance implications. Once they have defined the collection's structure, users can pull in data from any dataset, previously related or not. Then, if they want to

change the underlying datasets, users declare the new collection criteria. Redefining the collection structure or recompiling applications that access the collection is no longer necessary.

Prices for the System 1032 Version 9 product range from \$6,000 to \$240,000, depending on CPU size across the entire VAX family.

CompuServe Data Technologies, 1000 Massachusetts Ave., Cambridge, MA 02138, (617) 661-9440.

CIRCLE 104 ON READER SERVICE CARD

XDB LINKS AND SERVES

XDB Systems Inc. has released version 2.41 of XDB-Server, their SQL engine for OS/2, and XDB-Link, a new software package that lets PC applications access mainframe DB2 data transparently.

The XDB-Server is a high-performance, multiuser database system that brings the power and integrity of mainframe database systems to LANs. The core of the company's product line is the DB2 compatible Engine. With the XDB-Server for OS/2, DB2 applications become portable to client/server platforms. Since the XDB-Server is network independent, DB2 compatibility can exist on multiple platforms and operating systems.

XDB-Link helps users develop cooperative processing applications to use MIS resources more efficiently. For example, applications and data that reside in distributed areas can be viewed as a single data center.

The XDB-Link product consists of two components: XDB-Link Gateway, which resides on the PC, and the XDB-Link Host, which resides on the mainframe. These components communicate via LU6.2 protocols to satisfy DB2 SQL requests originated from the PC.

XDB-Link is based on a client/server architecture and supports DOS, OS/2, and Windows clients. In a development environment, the product extends systems-level testing to the PC environment by providing access to mainframe production data. In a production environment, XDB's application tools and LAN applications can now access data residing in main-

frame DB2 as a database server.

XDB-Server supports the Net-Bios and Named Pipes protocols and is priced at \$2,495. XDB-Link is priced from \$21,000 to \$36,000, depending on CPU size.

XDB Systems Inc., 14700 Sweitzer Ln., Laurel, MD 20707, (301) 317-6800.

CIRCLE 105 ON READER SERVICE CARD

MEET AT THE JUNCTION

Tools & Techniques Inc. has released Data Junction version 4.1, the company's data conversion tool. The product continues to exploit its "Hub & Spoke" architecture, letting users import and export their data files to and from virtually any file format. The product also extends the ability to sort, extract, rearrange, and edit records, fields, and bytes into the exact output format required. Parameters governing a Data Junction conversion can be selected interactively or set for automatic batch mode operation.

The product now supports memo field formats from popular databases. Other enhancements include improved support for various DATE formats; the introduction of "regular expressions" to perform Boolean extractions of records and search/replace of character strings; the ability to auto-align source and target fields by name; direct update of .dbf file structures at the record level; and scaling of numeric fields to assign decimal position.

DOS licenses for Data Junction 4.1 are priced at \$99, \$199, and \$299 for standard, professional, and advanced levels respectively. UNIX and LAN licenses start at \$400 for a four-user license.

Tools & Techniques Inc., 1620 W. 12th St., Austin, TX 78703, (512) 482-0824.

CIRCLE 106 ON READER SERVICE CARD

PLAY IT AGAIN, DB2SAM

Allen Systems Group Inc. has announced the release of DB2 Security Access Manager (DB2SAM), a software package that enhances DB2's security facilities and lets

sites organize and easily manage the DB2 security environment. This product also gives data processing organizations more control over DB2 security while delegating the tasks of individual "granting" and "revoking."

Through its ability to maintain and provide a comprehensive picture of all authorities and privileges, DB2SAM lets sites quickly locate and change any specific authorization entry. DB2SAM also helps sites keep control of security and prevent such situations as cascading DB2 authority. Other key features of the product include productivity in administering security, audit trails and reporting, and ease of security administration.

Prices for DB2SAM range from \$21,600 to \$31,700, depending on CPU configuration.

Allen Systems Group Inc., 750 11th St. S., Naples, FL 33940, (813) 263-6700.

CIRCLE 107 ON READER SERVICE CARD

STROBE LIGHTS THE WAY

Programart Corp. has announced the availability of Strobe Release 8.5, an application performance measurement software product for the IBM mainframe environment. Release 8.5, which incorporates new CICS, IDMS, COBOL, and PL/1 features, can help developers and database professionals improve application performance by rapidly pinpointing where and how applications spend their time.

According to the company, the new release significantly expands the depth of Strobe's ability to attribute resource consumption in language library and system service routines to specific parts of the application. Without this capability, programmers and technicians often waste time and resources trying to determine how application performance can be improved.

Strobe also measures the performance of online and batch processing applications. It takes "snapshots" of resource utilization while applications are running and produces performance profile reports, which detail specific areas of concentrated resource consumption. Used in each phase of the applica-

tion life cycle, Strobe can cut the cost of running applications, rapidly diagnose causes of performance crises, and maximize productivity.

The new Strobe IDMS feature includes three new reports that provide measurement and evaluation data to help application programmers and technicians identify inefficient or unnecessary use of system resources. Its new COBOL and PL/1 features include library attribution, making Strobe the first product capable of identifying the source program statements that implicitly call COBOL and PL/1 compiler library subroutines.

Prices range from \$18,000 to \$80,000, depending on CPU size.

Programart Corp., University Place, 124 Mt. Auburn St., Cambridge, MA 02138, (617) 661-3020.

CIRCLE 108 ON READER SERVICE CARD

EVERYONE'S TALKING NATURAL LANGUAGE

Natural Language Inc. has introduced Natural Language release 5.0, an English-based querying tool for relational databases that features an embedded, intuitive graphical user interface and an enhanced applications-development environment that reduces the deployment cycle. This release targets the desktop market by supporting most desktop UNIX machines. According to the company, the product lets users gain direct access to database information by asking for it in conversational English.

Release 5.0 features a menu-driven interface that complements Natural Language's ad hoc English access capabilities. Graphics, table manipulation, data analysis, and reports are now accessible through pull-down menus. A browsing capability offers a list of suggested questions that can be searched for by topic. Users can focus on the data they seek by browsing through the list of sample questions.

Other features of Natural Language 5.0 include online help that informs users about the product's capabilities at the time they need them, and a more sophisticated report generator that lets users manipulate tables and generate and save reports efficiently. The product also includes a file manager

that lets users integrate information with their most familiar tools, including Lotus's 1-2-3, Microsoft's Excel, and SAS.

Natural Language Inc. and Systems & Computer Technology Corp. (SCT) have also announced the release of IntelliQuest, SCT's English access querying tool based on the Natural Language product. IntelliQuest provides users with direct access to data in SCT's Banner Series of local government and higher education software products.

Pricing for Natural Language release 5.0 starts at \$4,500 for the lower-end desktop platforms. IntelliQuest is available from SCT, and pricing depends on the size of the institution or jurisdiction.

Natural Language Inc., 2910 Seventh St., Berkeley, CA 94710, (510) 841-3500.

Systems & Computer Technology Corp., 4 Country View Rd., Malvern, PA 19355, (215) 640-5137.

CIRCLE 109 ON READER SERVICE CARD

FORM A STRATEGIC INFOALLIANCE

Software Publishing Corp. has released InfoAlliance 1.10 for OS/2, a client/server software product designed to let PC users on a network transparently access and manipulate data located in disparate databases throughout the organization. The product contains all of the easy-to-use forms design, report writing, information access, and integration features of the originally introduced InfoAlliance for OS/2 Presentation Manager product. Additional capabilities in this OS/2 version include access to Microsoft SQL Server data, the ability to manipulate scanned images, support for OS/2 Dynamic Data Exchange, and 3GL triggers for programmers who wish to add custom features with C libraries.

A single InfoAlliance 1.10 standalone workstation product for either OS/2 or Windows clients is \$495. A workgroup installation with eight OS/2 or Windows clients and a single OS/2-based server is priced at \$6,455.

Software Publishing Corp., 3165 Kifer Rd., Santa Clara, CA 95051, (408) 450-7314.

CIRCLE 110 ON READER SERVICE CARD

All CASE vendors offer product. Only one offers the BACHMAN vision.

BACHMAN delivers
proven software
productivity through
Model Driven
Development™ now.

When you examine most CASE vendors, they end up looking very much alike.

All except for BACHMAN. We offer you a unique vision of how your organization can move into the next century of applications and systems development with products and services available now.

BACHMAN provides an architecture to dramatically increase productivity, supported by the best consulting and training services available.

Discover BACHMAN Model Driven Development.™

Start sharing in the vision today.

1-800-BACHMAN.

BACHMAN

CIRCLE 20 ON READER SERVICE CARD



Enough Said.

CA-IDMS® VS. DB2

	CA-IDMS	DB2
Industry Standards:		
Ansi SQL Support	✓	✓
FIPS	✓	✓
SAA	✓	✓
Ansi Standard Domain Integrity, Check Constraint	✓	
Portability: Platforms Supported...		
MVS: MVS/XA	✓	✓
MVS/ESA Dataspace	✓	✓
VSE	✓	
VSE/ESA	✓	
VM	✓	
PC-DOS	✓	
PC LAN	✓	
Fujitsu	✓	
Investment Protection: Runs Applications Without Change From...		
DL/I	✓	
IMS	✓	
VSAM	✓	✓
TOTAL	✓	
Thousands Of Mission-Critical Production Applications	✓	
Continuous Processing Through Dynamic Configuration Support	✓	
Industrial-Strength Application Development	✓	
SQL Access To Existing DB's	✓	
Integrated Relational SQL And Navigational Support In A Single DBMS	✓	
Client/Server Architecture	✓	✓
Century Dates	✓	✓
Integrated Dictionary	✓	
Transient Read	✓	
Referential Integrity	✓	✓
Row Level Locking	✓	
Roll-Back For Complete Backward Recovery	✓	
Case Tool Support	✓	
Bulk Access	✓	
Commit Continue	✓	
Pseudo-Conversational SQL	✓	
SQL Tuning Options: System-Maintained Clustering	✓	
Variable Page Sizes	✓	

When you line up all the facts, it's easy to see why CA-IDMS® is the leading, industrial-strength DBMS thousands and thousands of clients rely on.

Unlike DB2, CA-IDMS can handle all of your mission-critical applications.

It's supported by the most comprehensive architecture ever developed, CA90s: Computing Architecture for the 90s.



And it has all of the advanced features and functionality you need today.

So call 1-800-645-3003 to arrange for an on-site briefing or for more information.

Call now and get the facts.

As you can see, they speak for themselves.

COMPUTER ASSOCIATES®
Software superior by design.



Information Handling Services

**Fast, easy, cost-effective
information management...**



**puts the information
you need...**

**into a system
you can trust.**

VSMF[®] from Information Handling Services.

25
1959-1984

Excellence is our Standard

An **itg** Company — Information Technology Group

IHS stands alone in time and cost control of technical information searches.

**25 years of
information
management**

Information Handling Services was founded in 1959 by an engineer who was all too familiar with the problems of finding technical information and finding it fast. Since then, techniques developed by IHS have revolutionized the process of information handling. Engineering, purchasing and procurement professionals have come to trust the IHS name.

**Putting the customer
first has kept
us in the lead**

As the recognized leader in information handling, IHS serves more than 6,000 customers in more than 65 countries around the world. It is important to note that more than 90% of these customers renew our services annually. Simply put, we have been time tested by the toughest critics...our users.

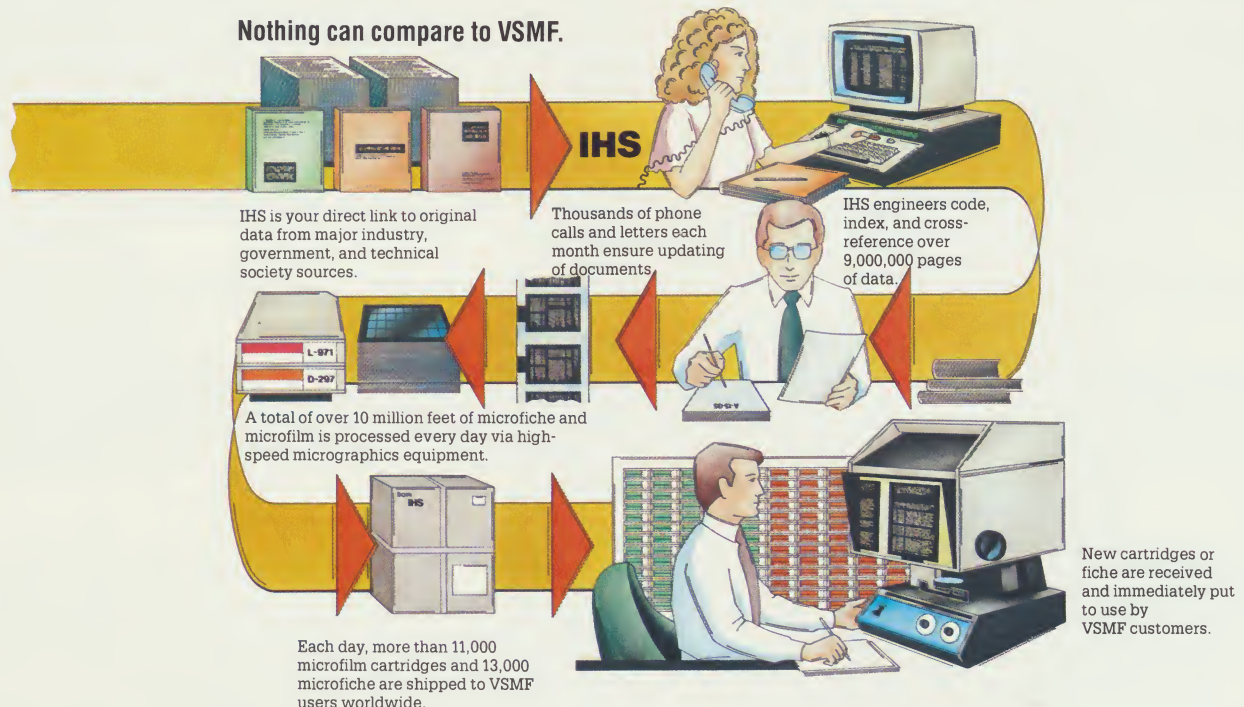
**Unequaled products and
services at competitive
prices bring you
the greatest value**

The world's largest source of vendor catalogs, industry standards, military and federal specifications, federal government and regulatory data is backed by the largest and most personalized service organization available. Add our component pricing structure and you'll realize a value that no other company can match.

**VSMF: the most
comprehensive technical
information system**

A value analysis of VSMF demonstrates unparalleled performance in acquiring, organizing and disseminating current information. For example, it includes more than 66,000 catalogs representing more than 26,000 carefully selected manufacturers.

Nothing can compare to VSMF.



For procurement specialists, it provides easy access to alternate sources and speeds the way to finding the best possible product at the best possible price. For engineers, it dramatically reduces search time resulting in increased productivity. The illustration on the preceding page will help explain the VSMF system.

VSMF is kept current with constant, regular updates

We know that your crucial procurement and design decisions depend on current information. So our staff of engineers and para-technicians work from nearly 200 CRT terminals linked to a high-performance data management system to track 9 million pages of current information.

VSMF is easy to use because of unique cross-referencing capabilities

The VSMF system allows you to quickly begin a search by product or subject description, vendor name, mil spec number, Qualified Products List (QPL), or other parameters. Through the use of either the printed index or TECH DATA™ online indexing, you can easily identify the appropriate cross-referencing code. This code will lead you immediately to related supplier catalogs, industry standards or government specifications you need to reference.

Exclusive cross-reference locator code makes the VSMF system easy to use.



Commercial BNC Dual-Crimp Plugs

FEATURES AND BENEFITS

- Low cost with quality crimp-on connectors at a smaller component budget
- Smaller diameter and lighter weight is perfect for compact designs
- Fully compatible for a uniform, high quality connector every time
- Capable center contact provides for secure, reliable connections
- RF performance equal to higher priced BNC connectors
- Fully compatible with all BNC UG-37 connectors to meet test needs without redesigning components
- Low VSWR for excellent performance
- Technical specifications increase connector life and reduce maintenance component appearance
- Ready and widely stocked with AMP hand tools for AMP Terminating Machine.

AMP COAXCON® Commercial BNC Dual-Crimp Plugs are the high cost of BNC connectors and reduce application tooling time required for termination. These small diameter coaxial connectors are quickly and easily crimped onto an AMP Terminating Machine, a CERTI CRIMP® hand tool, or the new commercial hand tool. The dual crimp operation secures both center and shield wires with the same tool. They give a uniform, secure connection without the problems of soldering. Performance is comparable to higher priced military connectors. The AMP Commercial Connector is fully interchangeable with all BNC UG-37 connectors.

AMP Commercial BNC Dual-Crimp Plugs may be used in virtually any BNC test installation. Performance is comparable with standard shield crimp. All Spec connectors are 0.500" dia. with the following dimensions counting from the shield to the 120° BNC interface, and

center contact resistance is slightly higher than standard.

The AMP Commercial BNC Connector is a fully interchangeable connector with the AMP Terminating Machine and the AMP CERTI CRIMP® hand tool.

Dimensions:

All dimensions in inches and millimeters. Values in brackets are metric equivalents.

1. 1/4" (6.35mm)

2. 1/4" (6.35mm)

3. 1/4" (6.35mm)

4. 1/4" (6.35mm)

5. 1/4" (6.35mm)

6. 1/4" (6.35mm)

7. 1/4" (6.35mm)

8. 1/4" (6.35mm)

9. 1/4" (6.35mm)

10. 1/4" (6.35mm)

11. 1/4" (6.35mm)

12. 1/4" (6.35mm)

13. 1/4" (6.35mm)

14. 1/4" (6.35mm)

15. 1/4" (6.35mm)

16. 1/4" (6.35mm)

17. 1/4" (6.35mm)

18. 1/4" (6.35mm)

19. 1/4" (6.35mm)

20. 1/4" (6.35mm)

21. 1/4" (6.35mm)

22. 1/4" (6.35mm)

23. 1/4" (6.35mm)

24. 1/4" (6.35mm)

25. 1/4" (6.35mm)

26. 1/4" (6.35mm)

27. 1/4" (6.35mm)

28. 1/4" (6.35mm)

29. 1/4" (6.35mm)

30. 1/4" (6.35mm)

31. 1/4" (6.35mm)

32. 1/4" (6.35mm)

33. 1/4" (6.35mm)

34. 1/4" (6.35mm)

35. 1/4" (6.35mm)

36. 1/4" (6.35mm)

37. 1/4" (6.35mm)

38. 1/4" (6.35mm)

39. 1/4" (6.35mm)

40. 1/4" (6.35mm)

41. 1/4" (6.35mm)

42. 1/4" (6.35mm)

43. 1/4" (6.35mm)

44. 1/4" (6.35mm)

45. 1/4" (6.35mm)

46. 1/4" (6.35mm)

47. 1/4" (6.35mm)

48. 1/4" (6.35mm)

49. 1/4" (6.35mm)

50. 1/4" (6.35mm)

51. 1/4" (6.35mm)

52. 1/4" (6.35mm)

53. 1/4" (6.35mm)

54. 1/4" (6.35mm)

55. 1/4" (6.35mm)

56. 1/4" (6.35mm)

57. 1/4" (6.35mm)

58. 1/4" (6.35mm)

59. 1/4" (6.35mm)

60. 1/4" (6.35mm)

61. 1/4" (6.35mm)

62. 1/4" (6.35mm)

63. 1/4" (6.35mm)

64. 1/4" (6.35mm)

65. 1/4" (6.35mm)

66. 1/4" (6.35mm)

67. 1/4" (6.35mm)

68. 1/4" (6.35mm)

69. 1/4" (6.35mm)

70. 1/4" (6.35mm)

71. 1/4" (6.35mm)

72. 1/4" (6.35mm)

73. 1/4" (6.35mm)

74. 1/4" (6.35mm)

75. 1/4" (6.35mm)

76. 1/4" (6.35mm)

77. 1/4" (6.35mm)

78. 1/4" (6.35mm)

79. 1/4" (6.35mm)

80. 1/4" (6.35mm)

81. 1/4" (6.35mm)

82. 1/4" (6.35mm)

83. 1/4" (6.35mm)

84. 1/4" (6.35mm)

85. 1/4" (6.35mm)

86. 1/4" (6.35mm)

87. 1/4" (6.35mm)

88. 1/4" (6.35mm)

89. 1/4" (6.35mm)

90. 1/4" (6.35mm)

91. 1/4" (6.35mm)

92. 1/4" (6.35mm)

93. 1/4" (6.35mm)

94. 1/4" (6.35mm)

95. 1/4" (6.35mm)

96. 1/4" (6.35mm)

97. 1/4" (6.35mm)

98. 1/4" (6.35mm)

99. 1/4" (6.35mm)

100. 1/4" (6.35mm)

101. 1/4" (6.35mm)

102. 1/4" (6.35mm)

103. 1/4" (6.35mm)

104. 1/4" (6.35mm)

105. 1/4" (6.35mm)

106. 1/4" (6.35mm)

107. 1/4" (6.35mm)

108. 1/4" (6.35mm)

109. 1/4" (6.35mm)

110. 1/4" (6.35mm)

111. 1/4" (6.35mm)

112. 1/4" (6.35mm)

113. 1/4" (6.35mm)

114. 1/4" (6.35mm)

115. 1/4" (6.35mm)

116. 1/4" (6.35mm)

117. 1/4" (6.35mm)

118. 1/4" (6.35mm)

119. 1/4" (6.35mm)

120. 1/4" (6.35mm)

121. 1/4" (6.35mm)

122. 1/4" (6.35mm)

123. 1/4" (6.35mm)

124. 1/4" (6.35mm)

125. 1/4" (6.35mm)

126. 1/4" (6.35mm)

127. 1/4" (6.35mm)

128. 1/4" (6.35mm)

129. 1/4" (6.35mm)

130. 1/4" (6.35mm)

131. 1/4" (6.35mm)

132. 1/4" (6.35mm)

133. 1/4" (6.35mm)

134. 1/4" (6.35mm)

135. 1/4" (6.35mm)

136. 1/4" (6.35mm)

137. 1/4" (6.35mm)

138. 1/4" (6.35mm)

139. 1/4" (6.35mm)

140. 1/4" (6.35mm)

141. 1/4" (6.35mm)

142. 1/4" (6.35mm)

143. 1/4" (6.35mm)

144. 1/4" (6.35mm)

145. 1/4" (6.35mm)

146. 1/4" (6.35mm)

147. 1/4" (6.35mm)

148. 1/4" (6.35mm)

149. 1/4" (6.35mm)

150. 1/4" (6.35mm)

151. 1/4" (6.35mm)

152. 1/4" (6.35mm)

153. 1/4" (6.35mm)

154. 1/4" (6.35mm)

155. 1/4" (6.35mm)

156. 1/4" (6.35mm)

157. 1/4" (6.35mm)

158. 1/4" (6.35mm)

159. 1/4" (6.35mm)

160. 1/4" (6.35mm)

161. 1/4" (6.35mm)

162. 1/4" (6.35mm)

163. 1/4" (6.35mm)

164. 1/4" (6.35mm)

165. 1/4" (6.35mm)

166. 1/4" (6.35mm)

167. 1/4" (6.35mm)

168. 1/4" (6.35mm)

169. 1/4" (6.35mm)

170. 1/4" (6.35mm)

171. 1/4" (6.35mm)

172. 1/4" (6.35mm)

173. 1/4" (6.35mm)

174. 1/4" (6.35mm)

175. 1/4" (6.35mm)

176. 1/4" (6.35mm)

177. 1/4" (6.35mm)

178. 1/4" (6.35mm)

179. 1/4" (6.35mm)

180. 1/4" (6.35mm)

181. 1/4" (6.35mm)

182. 1/4" (6.35mm)

183. 1/4" (6.35mm)

184. 1/4" (6.35mm)

185. 1/4" (6.35mm)

186. 1/4" (6.35mm)

187. 1/4" (6.35mm)

188. 1/4" (6.35mm)

189. 1/4" (6.35mm)

190. 1/4" (6.35mm)

191. 1/4" (6.35mm)

192. 1/4" (6.35mm)

193. 1/4" (6.35mm)

194. 1/4" (6.35mm)

195. 1/4" (6.35mm)

196. 1/4" (6.35mm)

197. 1/4" (6.35mm)

198. 1/4" (6.35mm)

199. 1/4" (6.35mm)

200. 1/4" (6.35mm)

201. 1/4" (6.35mm)

202. 1/4" (6.35mm)

203. 1/4" (6.35mm)

204. 1/4" (6.35mm)

205. 1/4" (6.35mm)

206. 1/4" (6.35mm)

207. 1/4" (6.35mm)

208. 1/4" (6.35mm)

209. 1/4" (6.35mm)

210. 1/4" (6.35mm)

211. 1/4" (6.35mm)

212. 1/4" (6.35mm)

213. 1/4" (6.35mm)

214. 1/4" (6.35mm)

215. 1/4" (6.35mm)

216. 1/4" (6.35mm)

217. 1/4" (6.35mm)

218. 1/4" (6.35mm)

219. 1/4" (6.35mm)

220. 1/4" (6.35mm)

221. 1/4" (6.35mm)

222. 1/4" (6.35mm)

223. 1/4" (6.35mm)

224. 1/4" (6.35mm)

225. 1/4" (6.35mm)

226. 1/4" (6.35mm)

227. 1/4" (6.35mm)

228. 1/4" (6.35mm)

229. 1/4" (6.35mm)

230. 1/4" (6.35mm)

231. 1/4" (6.35mm)

232. 1/4" (6.35mm)

233. 1/4" (6.35mm)

234. 1/4" (6.35mm)

235. 1/4" (6.35mm)

236. 1/4" (6.35mm)

237. 1/4" (6.35mm)

238. 1/4" (6.35mm)

239. 1/4" (6.35mm)

240. 1/4" (6.35mm)

241. 1/4" (6.35mm)

242. 1/4" (6.35mm)

243. 1/4" (6.35mm)

244. 1/4" (6.35mm)

245. 1/4" (6.35mm)

246. 1/4" (6.35mm)

247. 1/4" (6.35mm)

248. 1/4" (6.35mm)

249. 1/4" (6.35mm)

250. 1/4" (6.35mm)

251. 1/4" (6.35mm)

252. 1/4" (6.35mm)

253. 1/4" (6.35mm)

254. 1/4" (6.35mm)

255. 1/4" (6.35mm)

256. 1/4" (6.35mm)

257. 1/4" (6.35mm)

258. 1/4" (6.35mm)

259. 1/4" (6.35mm)

260. 1/4" (6.35mm)

261. 1/4" (6.35mm)

262. 1/4" (6.35mm)

263. 1/4" (6.35mm)

264. 1/4" (6.35mm)

265. 1/4" (6.35mm)

266. 1/4" (6.35mm)

267. 1/4" (6.35mm)

268. 1/4" (6.35mm)

269. 1/4" (6.35mm)

270. 1/4" (6.35mm)

271. 1/4" (6.35mm)

272. 1/4" (6.35mm)

273. 1/4" (6.35mm)

274. 1/4" (6.35mm)

275. 1/4" (6.35mm)

276. 1/4" (6.35mm)

277. 1/4" (6.35mm)

278. 1/4" (6.35mm)

279. 1/4" (6.35mm)

280. 1/4" (6.35mm)

281. 1/4" (6.35mm)

282. 1/4" (6.35mm)

283. 1/4" (6.35mm)

284. 1/4" (6.35mm)

285. 1/4" (6.35mm)

286. 1/4" (6.35mm)

287. 1/4" (6.35mm)

288. 1/4" (6.35mm)

289. 1/4" (6.35mm)

290. 1/4" (6.35mm)

291. 1/4" (6.35mm)

292. 1/4" (6.35mm)

293. 1/4" (6.35mm)

294. 1/4" (6.35mm)

295. 1/4" (6.35mm)

296. 1/4" (6.35mm)

297. 1/4" (6.35mm)

298. 1/4" (6.35mm)

299. 1/4" (6.35mm)

300. 1/4" (6.35mm)

301. 1/4" (6.35mm)

302. 1/4" (6.35mm)

303. 1/4" (6.35mm)

304. 1/4" (6.35mm)

305. 1/4" (6.35mm)

306. 1/4" (6.35mm)

307. 1/4" (6.35mm)

308. 1/4" (6.35mm)

309. 1/4" (6.35mm)

310. 1/4" (6.35mm)

311. 1/4" (6.35mm)

312. 1/4" (6.35mm)

313. 1/4" (6.35mm)

314. 1/4" (6.35mm)

315. 1/4" (6.35mm)

316. 1/4" (6.35mm)

317. 1/4" (6.35mm)

318. 1/4" (6.35mm)

319. 1/4" (6.35mm)

320. 1/4" (6.35mm)

321. 1/4" (6.35mm)

322. 1/4" (6.35mm)

323. 1/4" (6.35mm)

324. 1/4" (6.35mm)

325. 1/4" (6.35mm)

326. 1/4" (6.35mm)

327. 1/4" (6.35mm)

328. 1/4" (6.35mm)

329. 1/4" (6.35mm)

330. 1/4" (6.35mm)

331. 1/4" (6.35mm)

332. 1/4" (6.35mm)

333. 1/4" (6.35mm)

334. 1/4" (6.35mm)

335. 1/4" (6.35mm)

336. 1/4" (6.35mm)

337. 1/4" (6.35mm)

338. 1/4" (6.35mm)

339. 1/4" (6.35mm)

340. 1/4" (6.35mm)

341. 1/4" (6.35mm)

342. 1/4" (6.35mm)

343. 1/4" (6.35mm)

344. 1/4" (6.35mm)

345. 1/4" (6.35mm)

346. 1/4" (6.35mm)

347. 1/4" (6.35mm)

348. 1/4" (6.35mm)

349. 1/4" (6.35mm)

350. 1/4" (6.35mm)

351. 1/4" (6.35mm)

352. 1/4" (6.35mm)

353. 1/4" (6.35mm)

354. 1/4" (6.35mm)

355. 1/4" (6.35mm)

356. 1/4" (6.35mm)

357. 1/4" (6.35mm)

358. 1/4" (6.35mm)

359. 1/4" (6.35mm)

360. 1/4" (6.35mm)

361. 1/4" (6.35mm)

362. 1/4" (6.35mm)

363. 1/4" (6.35mm)

364. 1/4" (6.35mm)

365. 1/4" (6.35mm)

366. 1/4" (6.35mm)

367. 1/4" (6.35mm)

368. 1/4" (6.35mm)

369. 1/4" (6.35mm)

370. 1/4" (6.35mm)

371. 1/4" (6.35mm)

372. 1/4" (6.35mm)

373. 1/4" (6.35mm)

374. 1/4" (6.35mm)

375. 1/4" (6.35mm)

376. 1/4" (6.35mm)

377. 1/4" (6.35mm)

378. 1/4" (6.35mm)

379. 1/4" (6.35mm)

380. 1/4" (6.35mm)

381. 1/4" (6.35mm)

382. 1/4" (6.35mm)

383. 1/4" (6.35mm)

384. 1/4" (6.35mm)

385. 1/4" (6.35mm)

386. 1/4" (6.35mm)

387. 1/4" (6.35mm)

388. 1/4" (6.35mm)

389. 1/4" (6.35mm)

390. 1/4" (6.35mm)

391. 1/4" (6.35mm)

392. 1/4" (6.35mm)

393. 1/4" (6.35mm)

394. 1/4" (6.35mm)

395. 1/4" (6.35mm)

396. 1/4" (6.35mm)

397. 1/4" (6.35mm)

398. 1/4" (6.35mm)

399. 1/4" (6.35mm)

400. 1/4" (6.35mm)

401. 1/4" (6.35mm)

402. 1/4" (6.35mm)

403. 1/4" (6.35mm)

404. 1/4" (6.35mm)

405. 1/4" (6.35mm)

406. 1/4" (6.35mm)

407. 1/4" (6.35mm)

408. 1/4" (6.35mm)

409. 1/4" (6.35mm)

410. 1/4" (6.35mm)

411. 1/4" (6.35mm)

412. 1/4" (6.35mm)

413. 1/4" (6.35mm)

414. 1/4" (6.35mm)

415. 1/4" (6.35mm)

416. 1/4" (6.35mm)

417. 1/4" (6.35mm)

418. 1/4" (6.35mm)

419. 1/4" (6.35mm)

420. 1/4" (6.35mm)

421. 1/4" (6.35mm)

422. 1/4" (6.35mm)

423. 1/4" (6.35mm)

424. 1/4" (6.35mm)

425. 1/4" (6.35mm)

426. 1/4" (6.35mm)

427. 1/4" (6.35mm)

428. 1/4" (6.35mm)

429. 1/4" (6.35mm)

430. 1/4" (6.35mm)

431. 1/4" (6.35mm)

432. 1/4" (6.35mm)

433. 1/4" (6.35mm)

434. 1/4" (6.35mm)

435. 1/4" (6.35mm)

436. 1/4" (6.35mm)

437. 1/4" (6.35mm)

438. 1/4" (6.35mm)

439. 1/4" (6.35mm)

440. 1/4" (6.35mm)

441. 1/4" (6.35mm)

442. 1/4" (6.35mm)

443. 1/4" (6.35mm)

444. 1/4" (6.35mm)

445. 1/4" (6.35mm)

446. 1/4" (6.35mm)

447. 1/4" (6.35mm)

448. 1/4" (6.35mm)

449. 1/4" (6.35mm)

450. 1/4" (6.35mm)

451. 1/4" (6.35mm)

452. 1/4" (6.35mm)

453. 1/4" (6.35mm)

454. 1/4" (6.35mm)

455. 1/4" (6.35mm)

456. 1/4" (6.35mm)

457. 1/4" (6.35mm)

458. 1/4" (6.35mm)

459. 1/4" (6.35mm)

460. 1/4" (6.35mm)

461. 1/4" (6.35mm)

462. 1/4" (6.35mm)

463. 1/4" (6.35mm)

464. 1/4" (6.35mm)

465. 1/4" (6.35mm)

466. 1/4" (6.35mm)

467. 1/4" (6.35mm)

468. 1/4" (6.35mm)

469. 1/4" (6.35mm)

470. 1/4" (6.35mm)

471. 1/4" (6.35mm)

472. 1/4" (6.35mm)

473. 1/4" (6.35mm)

474. 1/4" (6.35mm)

475. 1/4" (6.35mm)

476. 1/4" (6.35mm)

477. 1/4" (6.35mm)

478. 1/4" (6.35mm)

479. 1/4" (6.35mm)

480. 1/4" (6.35mm)

481. 1/4" (6.35mm)

482. 1/4" (6.35mm)

483. 1/4" (6.35mm)

484. 1/4" (6.35mm)

485. 1/4" (6.35mm)

486. 1/4" (6.35mm)

487. 1/4" (6.35mm)

488. 1/4" (6.35mm)

489. 1/4" (6.35mm)

490. 1/4" (6.35mm)

491. 1/4" (6.35mm)

492. 1/4" (6.35mm)

493. 1/4" (6.35mm)

494. 1/4" (6.35mm)

495. 1/4" (6.35mm)

496. 1/4" (6.35mm)

497. 1/4" (6.35mm)

498. 1/4" (6.35mm)

499. 1/4" (6.35mm)

500. 1/4" (6.35mm)

501. 1/4" (6.35mm)

502. 1/4" (6.35mm)

503. 1/4" (6.35mm)

504. 1/4" (6.35mm)

505. 1/4" (6.35mm)

506. 1/4" (6.35mm)

507. 1/4" (6.35mm)

508. 1/4" (6.35mm)

509. 1/4" (6.35mm)

510. 1/4" (6.35mm)

511. 1/4" (6.35mm)

512. 1/4" (6.35mm)

513. 1/4" (6.35mm)

514. 1/4" (6.35mm)

515. 1/4" (6.35mm)

516. 1/4" (6.35mm)

517. 1/4" (6.35mm)

518. 1/4" (6.35mm)

519. 1/4" (6.35mm)

520. 1/4" (6.35mm)

521. 1/4" (6.35mm)

522. 1/4" (6.35mm)

523. 1/4" (6.35mm)

524. 1/4" (6.35mm)

525. 1/4" (6.35mm)

526. 1/4" (6.35mm)

527. 1/4" (6.35mm)

528. 1/4" (6.35mm)

529. 1/4" (6.35mm)

530. 1/4" (6.35mm)

531. 1/4" (6.35mm)

532. 1/4" (6.35mm)

533. 1/4" (6.35mm)

534. 1/4" (6.35mm)

535. 1/4" (6.35mm)

536. 1/4" (6.35mm)

537. 1/4" (6.35mm)

538. 1/4" (6.35mm)

539. 1/4" (6.35mm)

540. 1/4" (6.35mm)

541. 1/4" (6.35mm)

542. 1/4" (6.35mm)

543. 1/4" (6.35mm)

544. 1/4" (6.35mm)

545. 1/4" (6.35mm)

546. 1/4" (6.35mm)

547. 1/4" (

TECH DATA means less time searching for information, more time using it



Unlike other online search programs which require a computer-trained information specialist, TECH DATA allows professionals to "key" in requests for technical data themselves. The user can then go directly to the correct film location in the VSMF service—or even order hard copies online.

TECH DATA offers immediate access to 15 major technical and engineering databases. Industrial catalogs from more than 26,000 vendors. Over 90% of the most commonly referenced domestic and international industry codes and standards. The world's most comprehensive collection of unclassified U. S. military and federal documents. Bibliographic files such as NTIS, Inspec and Cincinnati Milacron's Robotics. Full-text databases such as the Kirk-Othmer Encyclopedia of Chemical Technology. And a great deal more.

VSMF makes more than 9 million pages of vital information manageable

The VSMF data base system is divided into five parts so there's no wasted time spent going through irrelevant information.

Part One is Vendor Catalogs

with over 26,000 manufacturers represented in more than 66,000 catalogs. 10,000,000 products and components are included. The information is presented in two formats: full catalog (cover-to-cover), or side-by-side presentation of like products to greatly simplify second sourcing and product comparison. The VSMF customer can choose either or both formats. Further, the customer can customize his VSMF system to include only those information segments that are required.

CAA
CCITT
CECC
CGA
CSA
DIN
ECMA
EIA
EPRI
EURODOC
GPA
IEC
IEEE
IES
IPC
ISA

ISO
JIS
MPTH
MSS
NACE
NEMA
NFPA
NFP(A)
RTCA
SAE
SASO
SBAC
SEMI
SMACNA
TAPPI
UL
And many more...

Part Two is Industry Standards

which includes more than 90% of the world's most commonly used domestic, international and foreign national standards.

AASHTO	AREA
AATCC	ARI
ACI	ARINC
AECMA	ASA
AFBMA	ASAE
A.G.A.	ASCE
AGMA	ASHRAE
AI	ASME
AIA	ASNT
AIA (NAS)	ASQC
AISC	ASSE
AICHe	ASTM
ANS	AWS
ANSI	AWWA
API	BSI

Part Three is Department of Defense Products

in what is acknowledged industry-wide as the world's largest collection of unclassified military specifications and standards. This information reflects more than 18 years of collecting important military documents.

Military Specifications and Standards
Military Handbooks
MS Drawings
NASA Standards
Naval Instructions and Directives
Qualified Products Lists
And much more...

Part Four is Regulatory Services

which has been developed to maintain awareness of, and help ensure compliance with constantly changing federal regulations. Information included in this database includes:

Material Safety Data Sheets
Final Safety Analysis Reports
EPA Reports
Railway R-1 Reports
Bioenvironmental Data

Part Five is Government Documents

which provides crucial information relating to federal agencies, government personnel requirements and government procurement schedules and regulations.

FAA
Federal Construction Regulations
Federal Specifications
VA
NRC
FPM
Contractor Catalogs and Price Lists
FSS Schedules
Government Acquisition Regulations

VSMF can increase engineers' productivity 20%

Microfilm proven more efficient than hard-copy files

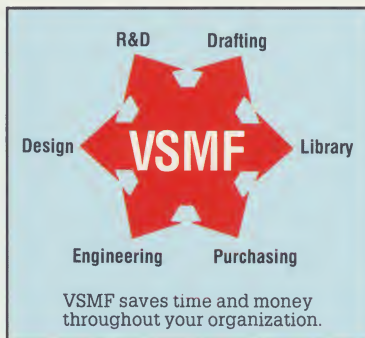


Traditional Search Time
49.5 minutes



VSMF Search Time
6 minutes

VSMF boosts efficiency organization wide, too



VSMF reduces leadtime problems

You don't have to buy more than you need, either

A recent survey* showed engineers spend an average of 49.5 minutes on each technical data search, for a weekly average of more than 7.4 hours. VSMF cuts average search time to 6 minutes per search or .9 hours each week. In the long run, VSMF can save as much as \$11,000 per year per engineer and boost productivity as much as 20%.

* Survey of 1,326 engineers. Search time based on an average of nine technical data searches each week. Cost savings based on average engineering hourly wage (with benefits and overhead) of \$34.50 and 50-week year. Hourly wage source: "Professional Income of Engineers"—American Association of Engineering Societies, 1983; average annual salary of \$34,500 for electronic engineers ten years out of college.

VSMF was designed by an engineer who wanted more time to be an engineer; greater flexibility in locating current data; and readable hard copies of drawings and specs from which to work. But the resulting system brings the same benefits to other professionals in the organization.

Anyone needing to procure, research, develop, design, manufacture or test products and systems according to the highest professional standards needs VSMF.

Inter-departmental use of VSMF assures standardized selection of parts and components and simplifies sourcing. So every department saves time and money and minimizes its error potential.

There's no need for purchasing and procurement managers or engineers to reinvent the wheel. Current information on alternate sources can affect important decisions on make vs. buy or determining acceptable levels of quality improvement. VSMF helps in finding the best possible product at the best price and delivery.

In fact, investing in VSMF can pay off other ways, too. By easing bottlenecks, speeding project completion, and cutting back on personnel requirements needed to maintain conventional data files.

Not every organization needs 9 million pages of technical data. The VSMF collection is divided into smaller specialized data segments to provide a custom-tailored VSMF program at an affordable price. Of course, your system can easily be expanded to meet future growth requirements.

Rely on the industry's most comprehensive Customer Service Network



Free instruction and support services via Extension 99



Person-to-Person Customer Service



As a VSMF user, you'll find out what hundreds of thousands of professionals already know: There is a Customer Service Network that you can call on at any time. Our Systems Service Representatives will install your VSMF data files, ensure the system is working properly and thoroughly train your staff and co-workers on an ongoing basis.

Your local System Service Representative (SSR) will regularly review your information needs to make sure your system is serving you most effectively. And, of course, your SSR will continue to train current and new personnel as needed.

We're as close as your phone. Although your staff will receive complete hands-on instruction for the best use of the VSMF system, difficult searches may require immediate assistance.

Whenever a technical search requires extra attention, a quick toll-free call to Extension 99 will put the caller in touch with our Customer Service Network and personal help.

Extension 99 offers direct contact with a technical information specialist who can guide the caller through a search step-by-step. Most problems are resolved immediately. The more difficult technical searches are solved by Extension 99 experts in no more than 24 hours.

In addition to Extension 99, we maintain a specially trained Customer Service staff to handle any questions or complaints any time you call. They will verify a shipping date, expedite replacement of broken or missing cartridges, lost or misplaced indexes. And, they'll help handle any repair problems with equipment leased from IHS.

IHS is committed to the customers that depend on our services. So we invest millions of dollars each year in personnel, training, service calls and systems to keep the IHS Customer Service Network the most reliable in the industry. No one else comes close.

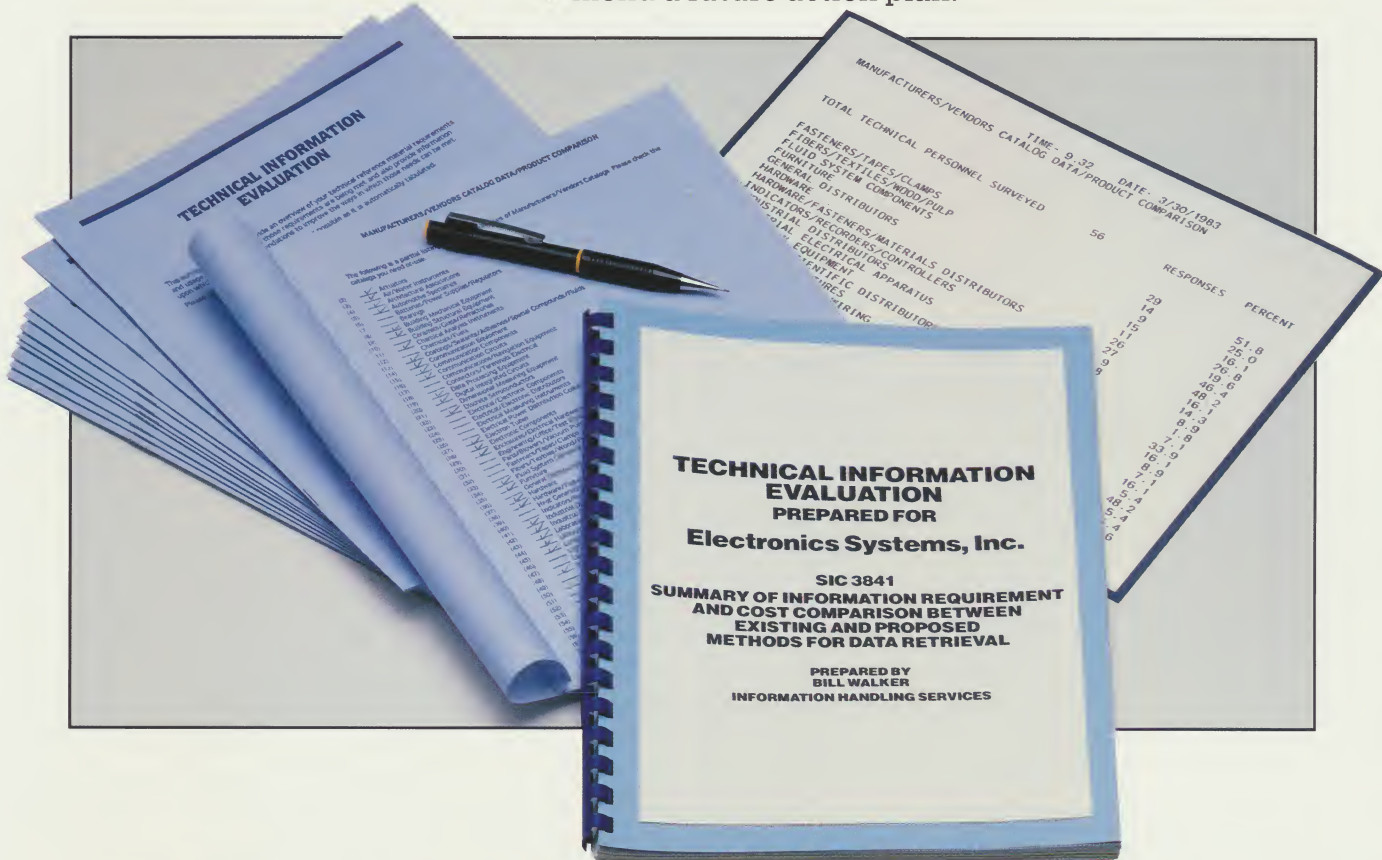
When we say "Excellence Is Our Standard," you can believe us. We know it's your standard, too.

An IHS exclusive... On-going Technical Information Evaluation

A value-added feature
you can't get
anywhere else,
and it's free

VSMF users can request a detailed, personalized analysis of their information requirements at no charge. The Technical Information Evaluation—valued at \$500—is a personally prepared confidential document for use solely by the customer and his Account Representative.

Computer generated, the Technical Information Evaluation will clearly and concisely identify problems and growth areas that need information search assistance. It will also record the time and money-saving benefits of VSMF and recommend a future action plan.



VSMF is the system of the future.

Your business isn't
standing still, and
neither is IHS

In the past year, we have informed our customers of over 30 important new refinements to the VSMF system. We added indexing features and sources, and refined presentation formats to provide necessary information more efficiently. New products were introduced to make sure current information needs were met. This constant refinement reflects the continuing commitment we have to the clientele which has made us the leader in the technical information retrieval industry. That's the only way to do business when excellence is your standard.

VSMF[®] saves time, saves money. Take their word for it.

Exxon Co.
General Motors Corp.
Mobil Corp.
**International Business
Machines Corp.**
General Electric Co.
Atlantic Richfield Co.
ITT Corp.
**R. J. Reynolds
Industries, Inc.**
Eastman Kodak Co.
The Dow Chemical Co.
Westinghouse Electric Corp.
The Boeing Co.
**The Goodyear Tire &
Rubber Co.**
Xerox Corp.
Pepsico, Inc.
Rockwell International Corp.
McDonnell Douglas Corp.
3M
The Coca-Cola Co.
W. R. Grace & Co.



Information Handling Services

15 Inverness Way East
Englewood, Colorado 80150
(800) 525-7052
(In Colorado, call 1-790-0600)
Excellence Is Our Standard.

"When our engineers and technicians were conducting searches manually, we estimated it took 40 hours per week and \$210,900 per year to find the data we needed. With VSMF it took 20 hours per week and only \$53,900. That's a savings of \$157,000 a year."

**Hughes Aircraft Company
Radar Systems Group**

"The supplier that supplied us a product last week had a unit price of \$2,200. We found another supplier in the IHS catalog that had the same product for exactly half the same price. By the time we had purchased all we needed, it saved us \$27,500."

Fort Gordon

"IHS' high standard of excellence is evidenced not only in the thorough refinement of new information products but in the professional demeanor and helpfulness of its representatives be it in person or by telephone through "Extension 99."

The VSMF system has clearly saved our Company time, space, and money in the research as well as the procurement functions by allowing our technically-oriented staff to view current industry-related standards as well as side-by-side comparisons of manufacturers' catalogs/products in an up-to-date and efficiently manageable format."

Florida Power & Light Company

